UNIVERSIDADE FEDERAL DE UBERLÂNDIA
INSTITUTO DE LETRAS E LINGUÍSTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTUDOS LINGUÍSTICOS

FERNANDO PAULINO DE OLIVEIRA

TOGATHERUP: UM PROTÓTIPO DE FERRAMENTA PARA A CONSTRUÇÃO DE *CORPORA*

UBERLÂNDIA
2019

FERNANDO PAULINO DE OLIVEIRA

TOGATHERUP: UM PROTÓTIPO DE FERRAMENTA DE COMPILAÇÃO DE *CORPUS*

Dissertação apresentada ao Programa de Pós-graduação em Estudos Linguísticos, do Instituto de Letras e Linguística, da Universidade Federal de Uberlândia como requisito parcial para obtenção do título de Mestre em Estudos Linguísticos.

Área de concentração: Estudos em Linguística e Linguística Aplicada.

Linha de pesquisa: Teoria, descrição e análise linguística.

Orientador: Prof. Dr. Guilherme Fromm

UBERLÂNDIA

2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

O48e
2019

Oliveira, Fernando Paulino de, 1981

ToGatherUp: um protótipo de ferramenta para a construção de *corpora* [recurso eletrônico] / Fernando Paulino de Oliveira. - 2019.

Orientador: Guilherme Fromm.

Dissertação (mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Estudos Linguísticos.

Modo de acesso: Internet.

Disponível em: http://dx.doi.org/10.14393/ufu.di.2019.679

Inclui bibliografia. Inclui ilustrações.

1. Linguística. 2. Linguística de corpus. 3. Língua inglesa - Computação. I. Fromm, Guilherme, 1968 - (Orient.) II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Estudos Linguísticos. III. Título.

CDU: 801

Gerlaine Araújo Silva - CRB-6/1408

FERNANDO PAULINO DE OLIVEIRA

TOGATHERUP: UM PROTÓTIPO DE FERRAMENTA PARA A CONSTRUÇÃO DE *CORPORA*

Dissertação apresentada ao Programa de Pós-graduação em Estudos Linguísticos, do Instituto de Letras e Linguística, da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Estudos Linguísticos.

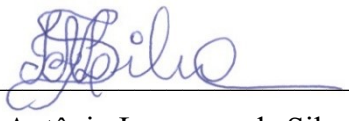
Área de concentração: Estudos em Linguística e Linguística Aplicada.

Linha de pesquisa: Teoria, descrição e análise linguística.

Uberlândia, 30 de maio de 2019.

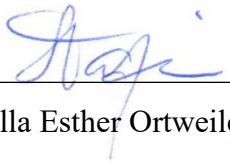
BANCA EXAMINADORA:

Prof. Dr. Guilherme Fromm (UFU)



Prof. Dr. Igor Antônio Lourenço da Silva (UFU)

Prof. Dr. Ariel Novodvorski (UFU)



Profa. Dra. Stella Esther Ortweiler Tagnin (USP)

Aos meus pais, Osimar e Erilda.

Aos meus irmãos, Osimar e Marcelo.

À minha esposa, Alinne.

Aos meus filhos, Santiago e Gabriel.

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Guilherme Fromm, por guiar-me pelos caminhos da pesquisa científica.

Agradeço aos membros da banca examinadora por dedicarem seu precioso tempo à leitura do meu trabalho.

Agradeço, especialmente, ao Prof. Dr. Ariel Novodvorski, que contribuiu de modo decisivo para os rumos desta pesquisa.

Agradeço à minha esposa Alinne por sempre apoiar-me e incentivar-me.

Agradeço ao pequeno Santiago que, por tantas vezes, concordou em me deixar estudar, mesmo sem entender direito o porquê.

Agradeço ao meu pai Osimar, que me ensinou a superar as dificuldades da vida.

Agradeço aos colegas do Grupo de Pesquisa e Estudos em Linguística de *Corpus* (GPELC), que compartilharam comigo seus conhecimentos e experiência.

Agradeço ao Muriel Ribeiro Alves por incentivar-me a criar o *ToGatherUp*.

Agradeço ao Heitor Carvalho de Almeida Neto, que manteve a eficiência da nossa equipe nos momentos em que estive ausente do trabalho.

Agradeço à Daniela Faria Grama, que revisou a escrita deste texto e apoiou-me desde a criação do meu projeto de pesquisa.

Agradeço, ainda, ao Programa de Pós-Graduação em Estudos Linguísticos (PPGEL), ao Instituto de Letras e Linguística (ILEEL) e à Universidade Federal de Uberlândia (UFU) pela oportunidade de crescimento acadêmico.

RESUMO

Esta pesquisa consiste em verificar o efeito da incorporação da ferramenta *ToGatherUp* no tempo e no esforço necessários para a construção manual de um *corpus* que elaboramos: o *Corpus* da Computação da Língua Inglesa (CoCLI). Para tanto, desenvolvemos um conjunto de métricas de medição de esforço – Esforço da Atividade (EA), Esforço Total de Coleta do Texto (ETCT) e Esforço Total do Projeto (ETP) – que serve de base para realizarmos um experimento estatístico comparativo entre os projetos de elaboração manual de duas versões idênticas do CoCLI que se diferenciam por em um deles utilizarmos o *ToGatherUp* e no outro não. A abordagem e a metodologia da Linguística de *Corpus*, em conjunto com os conceitos da área de Gerenciamento de Projetos, subsidiam a nossa proposta de sistematização do trabalho relativo à construção manual de *corpora*, a criação das duas versões do CoCLI e, juntamente com as noções da área da Computação, orientam-nos no desenvolvimento do *ToGatherUp*. O resultado do experimento demonstra uma redução média de 7,47% no ETP em que lançamos mão do *ToGatherUp* comparado ao ETP em que não utilizamos a ferramenta, o que corrobora a nossa hipótese de que ela reduz o tempo e o esforço despendidos pelo pesquisador em projetos de elaboração manual de *corpora*.

Palavras-chave: Linguística de *Corpus*. Construção manual de *corpus*. CoCLI. Métricas de medição de esforço. *ToGatherUp*.

ABSTRACT

This research verifies the effects of incorporating the *ToGatherUp* tool on both time and effort for building manually the corpus presented herein: the Corpus of Computing in English (CoCLI). We have developed a set of effort measurement metrics – Activity Effort (EA), Total Effort for Text Collection (ETCT) and Total Project Effort (ETP) – which served as the basis for conducting a comparative statistical experiment between the manual elaboration of two identical versions of the CoCLI: which differ from each other by one of them using the *ToGatherUp* and the other one not using it. The theory and methodology of Corpus Linguistics, together with the concepts from Project Management, subsidized our proposal of systematizing the manual construction of corpora and for creating the two versions of the CoCLI and, along with the notions of the Computing area, guided us in the development of *ToGatherUp*. The experiment shows an average reduction of 7.47% in the ETP when using *ToGatherUp* compared to the ETP when not using the tool. This result corroborates the hypothesis that the tool reduces the time and effort spent by the researcher on manual elaboration projects of corpora.

Keywords**:** *Corpus Linguistics. Manual construction of corpus.* CoCLI*. Effort measurement metrics. ToGatherUp.*

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACM	*Association for Computing Machinery*
AIS	*Association for Information Systems*
API	*Application Programming Interface*
ASCII	*American Standard Code for Information Interchange*
BNC	*British National Corpus*
BP	Banco do Português
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CCS	*Computing Classification System*
CE	*Computer Engineering*
CES	Câmara de Educação Superior
CNE	Conselho Nacional de Educação
COBUILD	*Collins Birmingham University International Language Database*
COCA	*Corpus of Contemporary American English*
CoCLI	*Corpus* da Computação da Língua Inglesa
CS	*Computer Science*
CS	*Computer Society*
CSS	*Cascading Style Sheets*
CSS	*Computing Classification System*
DARPA	*Defense Advanced Research Projects Agency*
DOC, DOCX	*Word Document* ou *Word Binary File Format*
EA	Esforço da Atividade
EACD	Esforço da Atividade de captura dos dados
EACT	Esforço da Atividade de cadastramento de textos
EACVD	Esforço da Atividade de conversão dos dados
EAED	Esforço da Atividade do enriquecimento dos dados
EALD	Esforço da Atividade da localização dos dados
EALND	Esforço da Atividade de limpeza e normalização dos dados
EANA	Esforço da Atividade de nomeação dos arquivos
EAOPD	Esforço da Atividade da obtenção de permissão de uso dos dados
EASA	Esforço da Atividade de salvamento de arquivos
ETCT	Esforço Total de Coleta do Texto
ETP	Esforço Total do Projeto
FAQ	*Frequently Asked Questions*
FNC	*File Naming Conventions*
GPELC	Grupo de Pesquisa e Estudos em Linguística de *Corpus*
HTML	*Hypertext Markup Language*
ID	Identificador
ILEEL	Instituto de Letras e Linguística
IS	*Information Systems*
ISO	*International Organization for Standardization*
IT	*Information Technology*
JSON	*JavaScript Object Notation*

KWIC	*Key Words in Context*
LC	Linguística de *Corpus*
LpC	Localização por Conteúdo
LpF	Localização por Formato
LS	Localizar e Substituir
MEC	Ministério da Educação
MIT	*Massachusetts Institute of Technology*
MW	*Microsoft Word*
MySQL	*My Structured Query Language*
OCR	*Optical Character Recognition*
ODT	*OpenDocument*
PC	*Personal Computer*
PDF	*Portable Document Format*
PHP	*Hypertext Preprocessor*
PLN	Processamento de Linguagem Natural
PMBOK	*Project Management Body of Knowledge*
POS	*Part-of-speech*
PPGEL	Programa de Pós-Graduação em Estudos Linguísticos
PUC	Pontifícia Universidade Católica
RDBMS	*Relational Database Management System*
RTF	*Rich Text Format*
SE	*Software Engineering*
SINALEL	Simpósio Nacional de Letras e Linguística
SGBD	Sistema Gerenciador de Bancos de Dados
SGML	*Standard Generalised Markup Language*
SQL	*Structured Query Language*
SSPS	*Statistics Statistical Package for the Social Sciences*
T-Factor	*ToGatherUp Effort Reduction Factor*
TCT	Teoria Comunicativa da Terminologia
TEI	*Text Encoding Initiative*
TXT	*Plain Text Format*
UFMG	Universidade Federal de Minas Gerais
UFPE	Universidade Federal de Pernambuco
UFRGS	Universidade Federal do Rio Grande do Sul
UFRJ	Universidade Federal do Rio de Janeiro
UFU	Universidade Federal de Uberlândia
UNICAMP	Universidade Estadual de Campinas
USP	Universidade de São Paulo
UTF-8	*8-bit Unicode Transformation Format*
VBA	*Visual Basic for Applications*
VoTec	Vocabulário Técnico Online
WST	*WordSmith Tools*
XML	*Extensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo, esclarecemos informações que consideramos primárias e, portanto, fundamentais à compreensão desta pesquisa. Em primeiro plano, apresentamos o autor desta dissertação de Mestrado; em segundo, justificamos a importância deste trabalho; em terceiro e quarto, expomos, respectivamente, a pergunta e a hipótese desta pesquisa; em quinto, pontuamos os objetivos dela e, em sexto, descrevemos como a organizamos.

1.1 Apresentação do autor

Meu[1] nome é Fernando Paulino de Oliveira e me formei em Letras[2], pela Universidade Federal de Uberlândia (UFU), em 2004. Logo após a conclusão da graduação, ingressei no serviço público, na carreira técnica-administrativa, e passei a atuar profissionalmente no Instituto de Letras e Linguística (ILEEL) da UFU. Em virtude de o ILEEL demandar serviços relacionados ao desenvolvimento de *sites* e sistemas com o fito de atender às necessidades acadêmicas e institucionais, comecei a participar de cursos de capacitação, a partir de 2010, com o intuito de adquirir os conhecimentos indispensáveis para a execução desses trabalhos. Como resultado disso, tornei-me um desenvolvedor de *sites* e sistemas.

Em 2015, decidi retomar os estudos relacionados à minha formação acadêmica (Letras) e pleitear uma vaga no curso de Mestrado do Programa de Pós-Graduação em Estudos Linguísticos (PPGEL) da UFU. Na época, de posse de um arcabouço de conhecimentos relativos à tecnologia, senti-me atraído e fascinado pela linha de pesquisa Teoria, descrição e análise linguística devido à proximidade dos estudos desenvolvidos nessa linha com a área da Computação.

Guiado pela motivação de tornar-me parte do universo dos estudos da linguagem que envolve tecnologias, em 2015, participei, como aluno, do curso de extensão Introdução aos Fundamentos da Linguística *Corpus*, promovido pelo Grupo de Pesquisa e Estudos em Linguística de *Corpus* (GPELC)[3]. Em 2016, fiz o curso de extensão Estatística para análise de

[1] Por apresentar informações pessoais, utilizo a primeira pessoa do singular nesta seção. Nas demais, adoto a primeira pessoa do plural.
[2] Licenciatura Plena em Inglês e Literaturas de Língua Inglesa.
[3] Grupo de pesquisa do ILEEL/UFU, criado em 2010 e coordenado, atualmente, pelos professores Dr. Guilherme Fromm e Dr. Ariel Novodvorski com o objetivo de desenvolver projetos relacionados à Linguística de *Corpus* e de reunir pesquisadores das áreas de Lexicografia, Lexicologia, Terminografia, Terminologia, Tradução, entre outras, que têm interesse em discutir estudos científicos que envolvam uso ou elaboração de *corpora*.

dados linguísticos, oferecido pelo ILEEL. Dessa forma, conheci os métodos da Linguística de *Corpus* (doravante LC) e obtive as noções que subsidiaram a elaboração do meu projeto de pesquisa na área da Terminologia/Terminografia.

Em 2017, ingressei no PPGEL da UFU e iniciei esta pesquisa com a expectativa de que minha experiência profissional na área de desenvolvimento de *softwares*, adquirida no hiato compreendido entre a conclusão da minha graduação e a entrada na pós-graduação, pudesse contribuir para os estudos relacionados à LC.

1.2 Contexto da pesquisa

No começo da nossa pesquisa, visávamos à produção de um vocabulário técnico-científico da área da Computação – uma obra terminográfica bilíngue (português/inglês), direcionada por *corpus*[4] e destinada aos alunos dos primeiros anos dos cursos da Computação. Para a realização desse trabalho, não previámos a necessidade de recorrermos aos especialistas da referida área, uma vez que, se os termos "são extraídos de *corpora* construídos de forma cuidadosa que reflitam a linguagem realmente usada no domínio que está sendo investigado, o produto é considerado altamente confiável e não depende necessariamente da validação de um especialista" (TAGNIN, 2012, p. 170).

Como base metodológica, seguíamos os princípios da LC que, conforme Berber Sardinha (2004, p. 3), "ocupa-se da coleta e da exploração de *corpora*, ou conjuntos de dados linguísticos textuais coletados criteriosamente com o propósito de servirem para a pesquisa de uma língua ou variedade linguística". No que diz respeito à fundamentação teórica, nos apoiávamos na Teoria Comunicativa da Terminologia (TCT), que foi criada por Maria Teresa Cabré e que tem como objeto de estudo o termo e seu caráter comunicativo nos ambientes específicos em que é utilizado.

Desse modo, o desenvolvimento da nossa pesquisa pressupunha a construção e a análise de dois *corpora*, um *corpus* da Computação em língua inglesa e outro em língua portuguesa. No entanto, dois problemas surgiram no processo de compilação e organização deles: o primeiro relaciona-se ao tamanho dos *corpora* e o segundo alude aos recursos tecnológicos disponíveis para lidarmos com eles.

[4] Definimos o termo *corpus* (plural *corpora*) neste capítulo, na seção 1.3 Justificativas e, mais adiante, no segundo capítulo, na seção 2.2 *Corpora*: conceitos e características.

De acordo com os critérios que havíamos estabelecido na etapa de planejamento da pesquisa, cada uma das áreas e subáreas da Árvore de Domínio da Computação[5] deveria conter, no mínimo, cerca de 100 mil palavras. Essa quantidade multiplicada pelo número de áreas e subáreas da Computação (2074 no total) gerou o primeiro obstáculo: ao final da compilação dos *corpora* da pesquisa, cada um deles possuiria, pelo menos, 207 milhões e 400 mil palavras (o correspondente à multiplicação das 2074 áreas e subáreas da Computação pelo número mínimo de 100 mil palavras).

Ao ponderarmos essa dimensão dos *corpora*, o tempo e o esforço necessários para a realização das atividades de construção[6] deles e os recursos humanos para o desenvolvimento desse trabalho, percebemos que ele era inexequível no Mestrado. Diante dessa percepção, fizemos um recorte das áreas e subáreas, o que resultou na manutenção de apenas 95 das 2074 áreas e subáreas iniciais. Essa redução tornou o projeto mais viável, já que os *corpora* passariam a ter nove milhões e 500 mil palavras (o correspondente à multiplicação de 95 áreas pelo número mínimo de 100 mil palavras).

Porém, ainda nessa etapa, deparamos com o segundo problema: o volume de textos[7] necessário para a construção dos dois *corpora* era muito grande e o uso dos métodos tradicionais de gerenciamento e armazenamento de arquivos (*softwares* genéricos, como o utilitário *Windows Explorer*, do sistema operacional *Windows*, e o *Excel*, da *Microsoft*) como suporte para as atividades da elaboração manual dos *corpora* parecia-nos ineficiente em face da possibilidade de utilizarmos métodos automatizados das tecnologias computacionais[8].

Assim, antes de iniciarmos a construção dos *corpora*, procuramos identificar, por meio de um levantamento[9], ferramentas já existentes da LC que pudessem nos auxiliar nas atividades de construção manual de *corpora* compostos por grande volume de dados[10]. Após a realização do levantamento, constatamos que as ferramentas disponíveis na atualidade não oferecem ou oferecem parcialmente recursos que podem nos ajudar. Tendo em vista a necessidade de solucionar essa questão em benefício do desenvolvimento da nossa própria

---

[5] Discutimos a distribuição de dados em áreas e subáreas e abordamos o conceito de Árvore de Domínio no segundo capítulo, na seção 2.2 *Corpora*: conceitos e características. As áreas e subáreas da Computação são apresentadas no terceiro capítulo, seção 3.3.2 Árvore de Domínio da Computação.

[6] Elencamos as atividades de construção de *corpora* no segundo capítulo, seção 2.4 O projeto de construção de *corpus*.

[7] Nesta pesquisa, utilizamos as expressões "volume de texto" e "volume de dados" como equivalentes.

[8] Nesta pesquisa, usamos: a) a expressão "tecnologia" de forma genérica para nos referirmos, alternativamente, de acordo com o contexto, aos *hardwares*, aos *softwares* e às técnicas computacionais; b) a expressão "tecnologias computacionais" como referência aos *softwares* e às técnicas computacionais; c) a expressão "ferramentas computacionais" como referência somente aos *softwares*.

[9] Apresentamos o levantamento neste capítulo, na seção 1.3 Justificativa.

[10] Nesta pesquisa, utilizamos "dados" para nos remetermos a "dados linguísticos" em formato textual e, também, aos textos de um *corpus.* Consideramos os textos como uma forma de apresentação de dados linguísticos.

pesquisa, criamos o *ToGatherUp*[11], que tem o objetivo de automatizar e facilitar o máximo possível as atividades da construção manual de *corpora.*

As dificuldades enfrentadas no início da pesquisa, em especial, a falta de ferramentas computacionais auxiliares no processo de elaboração de *corpora*, constituindo-se como uma grande lacuna, e a consequente criação do *ToGatherUp* fizeram com que o nosso foco de estudo mudasse. Passamos a nos interessar pelos efeitos da incorporação de tecnologias computacionais na construção manual de *corpora*.

Em decorrência disso, da proposta inicial da pesquisa (a construção de um vocabulário técnico-científico bilíngue da Computação), mantivemos somente a parte relacionada à construção de um *corpus* da Computação em língua inglesa. Elaboramos e utilizamos o *Corpus* da Computação da Língua Inglesa (doravante CoCLI) como fonte de dados para realizarmos um experimento e alcançarmos o objetivo desta pesquisa: determinar o efeito da incorporação do *ToGatherUp* no tempo e no esforço necessários para a execução de projetos de construção manual de *corpora*.

A abordagem que usamos para atingirmos tal objetivo consiste na realização de um experimento de comparação entre os esforços necessários para a construção de duas versões idênticas do CoCLI, sendo que no projeto de elaboração de uma delas utilizamos o *ToGatherUp* e no outro não. Para procedermos à confrontação, em um primeiro momento, percebemos a necessidade de estabelecermos um critério objetivo e um método para a medição do esforço das atividades de cada um dos projetos de construção de *corpora*. À medida que criamos os *corpora*, tabulamos os esforços necessários para a realização de cada uma das atividades dos projetos. Por fim, fizemos um teste estatístico para a comparação dos dados tabulados.

Desenvolvemos a pesquisa a partir do referencial teórico e metodológico da LC, da área de Gerenciamento de Projetos e da Computação. A teoria e a metodologia da LC, em conjunto com os conceitos de Gerenciamento de Projetos, subsidiam a nossa proposta de sistematização do trabalho de construção manual de *corpora* e a criação das duas versões do CoCLI. A Computação nos fornece fundamentos teóricos e metodológicos para a produção do *ToGatherUp* e para a realização das atividades dos projetos de construção do CoCLI.

[11] Disponível em: www.togatherup.ileel.ufu.br. Acesso em: 1 mar. 2019. Apresentamos o *ToGatherUp* no terceiro capítulo, na seção 3.2 O *ToGatherUp*.

1.3 Justificativa

A introdução dos computadores e de equipamentos, como os digitalizadores de texto (*scanners*) e de *softwares* de Reconhecimento Ótico de Caracteres (*Optical Character Recognition* – OCR), no fazer linguístico permitiu a criação de textos em formato eletrônico e a transformação de textos registrados em meios físicos (impressos ou manuscritos) para o formato eletrônico. O novo estado dos dados linguísticos possibilitou a elaboração de *corpora*, ou seja, conjuntos de dados linguísticos em formato eletrônico, provenientes de situações comunicativas reais e agrupados de forma criteriosa "com o propósito de servirem para a pesquisa de uma língua ou variedade linguística" (BERBER SARDINHA, 2004, p. 3).

Os *corpora* têm sido adotados como fontes de pesquisa em diferentes subáreas da Linguística, por exemplo, na Tradução, na Lexicologia, na Lexicografia, na Terminologia e na Terminografia, além de também estarem presentes em estudos de outras áreas, como na Linguística Computacional e na Literatura.

Na Tradução, de acordo com Tagnin (2015), os *corpora* têm sido utilizados em investigações que envolvem a comparação de textos originais e suas traduções para buscar equivalentes (2015, p. 38), a verificação de possibilidades de tradução (2015, p. 45), por exemplo, no que diz respeito à tradução de nomes próprios, palavras estrangeiras e termos culturalmente marcados (2015, p. 46), a confrontação de diferentes versões de uma tradução (2015, p. 49) e a revisão de textos traduzidos (2015, p. 50). Para a referida autora, os *corpora* possuem a vantagem de permitirem que estudantes e profissionais estendam "suas buscas por soluções de tradução para além dos limites das fontes dicionarizadas" (TAGNIN, 2015, p. 9).

Na Lexicologia e na Terminologia, é comum a obtenção de listas de palavras ordenadas de acordo com a frequência delas nos *corpora*. Os elementos das listas de palavras são usados como referência para a criação de linhas de concordância ou *Key Words in Context* (*KWIC*) – ocorrências de um elemento particular nos *corpora* acompanhadas do contexto linguístico em que foi empregado (BAKER; HARDIE; MCENERY, 2006) – no intuito de possibilitar ao pesquisador o estudo do léxico de uma língua, a descoberta de padrões linguísticos ou a identificação dos termos de uma área do conhecimento. Geralmente, as listas de palavras e as linhas de concordância são escrutinadas por lexicógrafos e terminógrafos a fim de granjearem, por exemplo, acepções e exemplos de uso de palavras e termos para a criação de dicionários.

Na Linguística Computacional, os *corpora* são etiquetados[12] e utilizados na identificação de "modelos linguísticos[13]" para treinamento de algoritmos de Processamento de Linguagem Natural (PLN) (PUSTEJOVSKY; STUBBS, 2012, p. 13). Na Literatura, os *corpora* embasam pesquisas que analisam questões estilísticas relacionadas, por exemplo, à autoria, dicção e caracterização no contexto de obras isoladas ou de um universo de obras de um mesmo autor (HOOVER; CULPEPER; O'HALLORAN, 2014) ou de vários autores.

O desenvolvimento de pesquisas com base na observação empírica de dados da língua favoreceu o surgimento e o crescimento da LC, que é "uma nova metodologia (que utiliza textos naturais e ferramentas informáticas para descrever a língua) e uma nova disciplina (no sentido de uma nova abordagem à descrição linguística)" (FRANKENBERG-GARCIA, 2012, p. 12). Podemos dizer que ela oferece os critérios, os pressupostos teóricos e metodológicos e um conjunto de ferramentas computacionais ao pesquisador para o trabalho com *corpora*. Apesar desse arsenal, nesta pesquisa, identificamos uma lacuna em relação à disponibilidade de ferramentas computacionais direcionadas a projetos de construção manual de *corpora* compostos por grandes volumes de dados, situação que esclarecemos nos próximos parágrafos.

Conforme Berber Sardinha (2004), para que seja possível o uso prático da LC, o interessado precisa de "um ingrediente essencial: o *corpus*" (BERBER SARDINHA, 2004, p. 45). Na atualidade, podemos encontrar uma grande quantidade de *corpora* prontos disponíveis em repositórios na Internet, como o Projeto Comet (http://comet.fflch.usp.br) e o BYU *corpora* (https://www.english-corpora.org), que podem ser adotados em trabalhos científicos. No entanto, considerando que os *corpora* são construídos de acordo com objetivos diferentes, a identificação de um *corpus* já existente que se adeque às necessidades de uma pesquisa pode não ser factível e o pesquisador pode ser compelido a elaborar o próprio *corpus* de estudo dele.

A construção de *corpora* de pequenas extensões[14] pode não representar um desafio, mas a de *corpora* compostos por grande volume de dados tem sido reportada como uma das partes mais difíceis do desenvolvimento de uma pesquisa (cf. ATKINS; CLEAR; OSTLER, 1992, p. 4; MACMULLEN, 2003, p. 15; SEMINO; SHORT, 2004, p. 226; MCENERY; XIAO; TONO, 2006, n. p; RENOUF, 2007, p. 42; EVANS, 2007, n. p; VOORMANN; GUT,

[12] Ver, no segundo capítulo, o tópico 2.4.2.2 O processo de preparação dos dados do *corpus*.
[13] De acordo com Neto (1980), um modelo linguístico é um construto matemático análogo à estrutura que subjaz aos dados linguísticos.
[14] A extensão ou o tamanho de um *corpus* representa o volume de dados linguísticos disponíveis para análise. No segundo capítulo, no tópico 2.2 *Corpora*: conceitos e características, discutimos sobre a extensão de *corpora*.

2008, p. 237; BAKER, 2010, p. 109; KÜBLER; ASTON, 2010, p. 512; MCENERY; HARDIE, 2011, p. 4; BIANCHI, 2012, p. 36; MINSHALL, 2013, p. 20; ZANETTIN, 2014, p. 32; EDWARD, 2015, p. 36). A principal reclamação dos linguistas refere-se à quantidade enorme de tempo e esforço necessária para a realização das atividades relativas à construção de *corpora* que demandam a constante intervenção manual[15] de acordo com a proporção do volume de dados de um *corpus*.

As dificuldades de elaboração de *corpora* podem ser contornadas pela facilidade oferecida por ferramentas computacionais que fazem a coleta automática de dados na Internet, como o *WebBootCat* (BARONI *et al.*, 2006), o *WebCorp Linguist's Search Engine* (KEHOE; GEE, 2007) e o *Bootcat* (BARONI; BERNARDINI, 2004). Entretanto, os *corpora* provenientes de dados coletados de forma automática apresentam graves problemas[16] quanto aos critérios de recuperação de dados utilizados pelas ferramentas computacionais, aos critérios de composição da amostra representada pelo *corpus*, à interferência causada pelo "ruído linguístico" na tokenização e nos cálculos estatísticos realizados durante o processamento do *corpus* para a criação das listas de palavras. Devido a esse quadro, parte dos linguistas desconsidera a utilização desses tipos de *corpora* em suas pesquisas.

A indisponibilidade de *corpora* prontos adequados às necessidades da pesquisa e o descarte da utilização de *corpora* produzidos de forma automática guiam o pesquisador para o caminho da construção manual de *corpora*. Para Edward (2015) e Garretson (2008), ao começar essa tarefa, uma das primeiras barreiras enfrentadas pelo pesquisador é encontrar ferramentas computacionais desenhadas especificamente para dar suporte[17] especializado às atividades do projeto. A ausência de *softwares* para tal finalidade leva-o a adotar recursos computacionais genéricos, como o utilitário *Windows Explorer*, do sistema operacional *Windows*, e o *Excel*, da *Microsoft*, para o auxiliarem nas atividades que dizem respeito à formação dos *corpora* de estudo.

Segundo Garretson (2008), lançar mão desses tipos de ferramenta nunca satisfaz realmente as necessidades específicas de um projeto. Concordamos com o autor e

[15] No segundo capítulo, na seção 2.5 O tempo e o esforço na construção de um *corpus*, discorremos sobre as necessidades de intervenção manual nas atividades relacionadas à construção de *corpora*.

[16] No mesmo capítulo e seção, apresentamos uma discussão detalhada e uma descrição extensiva acerca das dificuldades existentes na construção de *corpora* a partir de dados coletados de forma automática. Em razão dos problemas da coleta automática de dados, optamos por delimitar o foco da nossa pesquisa somente aos aspectos que aludem à compilação manual de *corpora*.

[17] Do nosso ponto de vista, as ferramentas que oferecem suporte à construção manual de *corpora* são aquelas que oferecem recursos que facilitam as atividades e o gerenciamento do projeto de construção manual de *corpora*. A facilitação das atividades pode se dar pela automatização (de acordo com critérios definidos pelo criador do *corpus*) de tarefas repetitivas e pela disponibilização de informações sobre a coleta de dados.

acrescentamos que, do nosso ponto de vista, o uso de recursos genéricos, apesar de ter funcionado como suporte para as demandas de diversas pesquisas, é improdutivo por exigir uma grande intervenção manual do pesquisador (suscetível a erros, dependente da repetição desnecessária de ações e, por vezes, lenta). Acreditamos, ainda, que optar por esses recursos é, em certa medida, deixar de contemplar a possibilidade de criação de ferramentas especializadas para a construção manual de *corpora* que, ao incorporarem a automatização de tarefas e a disponibilização de informações sobre a coleta de dados, podem facilitar as atividades do projeto, reduzindo o tempo e o esforço do pesquisador para a sua completude.

A partir das colocações de Edward (2015) e Garretson (2008) sobre a indisponibilidade de ferramentas e das nossas observações, nos questionamos se a LC apresenta, na atualidade, ferramentas computacionais que ofereçam o suporte (cf. nota de rodapé 17) às atividades de construção manual de *corpora* compostos por grandes volumes de dados. Para respondermos à nossa pergunta, analisamos dez ferramentas[18] da LC apontadas para a criação de *corpora* pelo projeto *Corpus Analysis*[19] (KLEIBER; BERBERICH, 2018), desenvolvido por Ingo Kleiber e Kristin Berberich, da Universidade de Heidelberg, na Alemanha.

Verificamos: o *AntCorGen* (ANTHONY, 2018)[20], o *AntFileSplitter* (ANTHONY, 2015)[21], o *BootCat* (BARONI; BERNARDINI, 2004)[22], o *CLaRK* (SIMOV *et al*., 2001)[23], o *ICEweb* (WEISSER, 2008)[24], o *NoSketch Engine* (RYCHLÝ, 2007)[25], o *SketchEngine* (KILGARRIFF; RYCHLÝ, 2004)[26], o *Sub-Corpus Creator* (LIANG; JIAJIN, 2011)[27], o *TextDirectory* (KLEIBER, 2018) [28] e o *TextSTAT* (HÜNING, 2014)[29].

Efetuamos essa análise por meio do UltraLex (FROMM; VICTOR, 2018)[30], que é uma ferramenta de avaliação de programas voltados para Lexicografia, Terminografia e Onomástica, criada por Guilherme Fromm e Samuel Victor, da UFU, em Uberlândia. No UltraLex, geramos um relatório para cada uma das ferramentas (cf. **APÊNDICE A –**

[18] As ferramentas estão classificadas nas categorias *corpus creation* e *compilation* no projeto *Corpus Analysis*.
[19] O *Corpus Analysis* apresenta uma relação extensiva das ferramentas disponíveis no mercado para o trabalho com *corpora*.
[20] Disponível em: http://www.laurenceanthony.net/software. Acesso em: 5 dez. 2018.
[21] Disponível em: http://www.laurenceanthony.net. Acesso em: 5 dez. 2018.
[22] Disponível em: http://bootcat.dipintra.it. Acesso em: 5 dez. 2018.
[23] Disponível em: http://bultreebank.org/en/clark/. Acesso em: 5 dez. 2018.
[24] Disponível em: http://martinweisser.org/ling_soft.html. Acesso em: 5 dez. 2018.
[25] Disponível em: https://nlp.fi.muni.cz/trac/noske. Acesso em: 5 dez. 2018.
[26] Disponível em: http://www.sketchengine.eu. Acesso em: 5 dez. 2018.
[27] Disponível em: http://corpus.bfsu.edu.cn/tools. Acesso em: 5 dez. 2018.
[28] Disponível em: https://github.com/IngoKl/textdirectory. Acesso em: 5 dez. 2018.
[29] Disponível em: http://neon.niederlandistik.fu-berlin.de/de/textstat/. Acesso em: 5 dez. 2018.
[30] Disponível em: http://ultralex.ileel.ufu.br. Acesso em: 5 dez. 2018.

Relatórios UltraLex), identificando as características e funcionalidades delas e tecendo observações sobre elas. Vale ressaltar que focamos principalmente nas funcionalidades de suporte à construção manual de *corpora* descritas no Quadro 1.

Quadro 1 – Funcionalidades de suporte à construção manual de *corpora* do UltraLex

Funcionalidades	Descrição
a) Armazenamento automático e customizado de dados	indica se a ferramenta faz o armazenamento automático dos dados do *corpus* de acordo com hierarquia de arquivos definida pelo criador do *corpus*.
b) Configuração do esquema de armazenamento de dados	indica se a ferramenta permite a configuração do armazenamento conforme hierarquia de arquivos definida pelo criador do *corpus*.
c) Controle de tempo investido na coleta de texto	indica se a ferramenta possui opção de registro de tempo utilizado na coleta de um texto.
d) Controle de tempo gasto na construção do *corpus*	indica se a ferramenta exibe o total de tempo gasto no projeto de construção do *corpus.*
e) Conversão de dados para formato TXT	indica se a ferramenta oferece suporte para a conversão automática de arquivos para o formato TXT.
f) Conversão de dados para o padrão *Unicode UTF-8*	indica se a ferramenta oferece suporte para a conversão automática de arquivos para o padrão *Unicode UTF-8*.
g) Estatísticas da coleta de dados do c*orpus*	indica se a ferramenta exibe estatísticas sobre a coleta de dados do *corpus*.
h) Etiquetagem manual do c*orpus*	indica se a ferramenta oferece suporte para a realização de etiquetagem manual do *corpus*.
i) Exportação customizada dos dados do c*orpus*	indica se a ferramenta exporta o *corpus* ou permite o *download* dele de acordo com hierarquia de arquivos definida pelo criador do *corpus*.
j) Inclusão, exclusão e edição de dados do *corpus*	indica se a ferramenta permite a manipulação básica dos dados do *corpus*.
k) Inserção automática de cabeçalhos com metadados	indica se a ferramenta faz a inserção de cabeçalhos com metadados dos textos nos arquivos do *corpus* de forma automática.
l) Nomeação automática e convencionada de arquivos	indica se a ferramenta permite ao criador do *corpus* convencionar a nomeação de arquivos e se ela executa a nomeação automática de acordo com a convenção estabelecida pelo criador.

m) Pesquisa de textos com base em metadados	indica se a ferramenta permite a recuperação/filtragem de textos por meio dos metadados deles.
n) Relatório/Listagem de dados coletados	indica se a ferramenta exibe a relação dos textos do *corpus* com os metadados deles.

Fonte: o autor.

A seguir, no Quadro 2, resumimos a nossa análise referente às funcionalidades de suporte à construção manual de *corpora*, identificando quais ferramentas as incorporam.

Quadro 2 – Funcionalidades de suporte à construção manual de *corpora* presentes nas ferramentas da LC

Funcionalidades	Ant Cor Gen	AntFile Splitter	Boot Cat	CLaRK	ICE web	No Sketch Engine	Sketch Engine	Sub-Corpus Creator	Text Directory	Text STAT
a) Armazenamento automático e customizado de dados	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
b) Configuração de esquema de armazenamento de dados	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
c) Controle de tempo gasto na coleta de texto	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
d) Controle de tempo gasto na construção do *Corpus*	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
e) Conversão de dados para formato TXT	Sim	Não	Sim	Não	Sim	Não	Não	Não	Não	Sim
f) Conversão de dados para padrão *Unicode UTF-8*	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
g) Estatísticas sobre a coleta de dados do *corpus*	Não	Não	Não	Não	Não	Não	Sim	Não	Não	Não
h) Etiquetagem manual do *corpus*	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
i) Exportação customizada dos dados do *corpus*	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
j) Inclusão, exclusão e edição de dados do *corpus*	Não	Não	Não	Não	Não	Não	Sim	Não	Não	Não
k) Inserção automática de cabeçalhos com metadados	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
l) Nomeação automática e convencionada de arquivos	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
m) Pesquisa de textos com base em metadados	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
n) Relatório/Listagem de dados coletados	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não

Fonte: o autor.

Conforme podemos ver no Quadro 2, o resultado da análise revelou que as ferramentas atuais da LC não oferecem ou oferecem parcialmente o suporte às atividades de elaboração manual de *corpora*. Isso corrobora a lacuna mencionada anteriormente e abre espaço para uma das contribuições da nossa pesquisa: a construção do *ToGatherUp* – uma ferramenta computacional que dá suporte às atividades de projetos que exijam a elaboração manual de *corpora* com grande volume de dados, capaz de automatizar tarefas, como a nomeação de arquivos, a inserção de cabeçalhos com metadados e o armazenamento dos dados do *corpus*, e de ainda apresentar informações sobre a coleta de dados que contribuam para o gerenciamento do projeto.

1.4 Pergunta de Pesquisa

A criação do *ToGatherUp* suscitou outro questionamento que se consolidou como pergunta de pesquisa: qual o efeito da incorporação de uma tecnologia computacional (no caso, o *ToGatherUp*) no tempo e no esforço necessários para a realização da construção manual de um *corpus* de grandes proporções (composto por grandes volumes de dados)?

1.5 Hipótese

Nossa hipótese é a de que a incorporação do *ToGatherUp* em projetos de construção manual de *corpora* poupa o tempo e minimiza o esforço do pesquisador dispensados à execução das atividades de elaboração de *corpora*, de modo semelhante ao que ocorre com as atividades de análise de *corpora* mediadas pelo uso de computadores (criação automática de listas de palavras e linhas de concordância, evidenciação de padrões linguísticos e etiquetagem de *corpora*).

1. 6 Objetivos geral e específicos

Nesta seção, apresentamos os objetivos geral e específicos desta pesquisa.

1.6.1 Objetivo geral

O objetivo geral é determinar o efeito da incorporação do *ToGatherUp* no tempo e no esforço necessários para a realização da construção manual do CoCLI.

1.6.2 Objetivos específicos

Os objetivos específicos da pesquisa são:

1. Realizar um levantamento das ferramentas já existentes da LC que oferecem suporte às atividades de construção manual de *corpora*;
2. Criar o *ToGatherUp*;
3. Construir uma versão do CoCLI com o uso do *ToGatherUp* e outra idêntica sem o *ToGatherUp*;
4. Elaborar uma proposta de sistematização do trabalho relativo à construção manual de *corpora* a partir dos princípios da LC e da área de Gerenciamento de Projetos;
5. Desenvolver um critério e um método para a medição do esforço despendido nas atividades de construção de *corpora*;

Comparar, por meio de um experimento, o esforço investido na construção do CoCLI com o *ToGatherUp* com o esforço empregado na elaboração do mesmo *corpus* sem o *ToGatherUp*.

1.7 Organização da dissertação

Organizamos esta dissertação em capítulos. No primeiro, que é este, apresentamos o autor desta pesquisa e discorremos sobre o contexto, a justificativa, a pergunta, a hipótese e os objetivos dela. No segundo, tratamos do referencial teórico e metodológico em que nos apoiamos. No início dele, abordamos a influência das tecnologias computacionais na produção de dicionários; em seguida, apresentamos os fundamentos da LC e da área de Gerenciamento de Projetos, pensando em uma proposta de sistematização do trabalho relativo à construção manual de *corpora*; e, no final, versamos sobre os referenciais da LC que abordam o tempo e o esforço na construção de um *corpus.* No terceiro, descrevemos a metodologia desta pesquisa. Em um primeiro

momento, mencionamos como estabelecemos um critério objetivo e um método para a medição do esforço das atividades de cada um dos projetos de construção do CoCLI. Na sequência, detalhamos como construímos as duas versões do CoCLI e os instrumentos que utilizamos para a coleta de dados que fizeram parte da análise do experimento da pesquisa. No quarto, expomos e discutimos os resultados do experimento. No quinto e último capítulo, tecemos as nossas considerações finais.

2 REFERENCIAL TEÓRICO E METODOLÓGICO

Neste capítulo, apresentamos o referencial teórico e metodológico em que nos apoiamos. No tópico 2.1, abordamos a influência das tecnologias computacionais na elaboração de dicionários e discorremos sobre como as inovações tecnológicas, ao longo do tempo, exerceram um papel determinante na definição das práticas dessa produção e na emergência e consolidação da LC. Em seguida, nos tópicos 2.2, 2.3 e 2.4, apresentamos os fundamentos da LC e da área de Gerenciamento de Projetos, pensando em uma proposta de sistematização do trabalho relativo à construção manual de *corpora*. Por fim, no tópico 2.5, versamos sobre os referenciais da LC que abordam o tempo e o esforço na construção de um *corpus*.

2.1 A influência das tecnologias computacionais na produção de dicionários

O surgimento dos computadores e o progresso das tecnologias computacionais provocaram uma grande transformação na relação entre as áreas de conhecimento e seus objetos de estudo. As subáreas da Linguística dedicadas à elaboração de dicionários, a Lexicografia e a Terminografia, são campos em que a influência das tecnologias computacionais de armazenagem, processamento e recuperação de informações impulsionou o desenvolvimento de novos métodos de trabalho.

Para Rundell e Kilgarriff (2011), a relação entre os dicionários e os computadores teve início na década de 1960. Nesse período, o uso dos computadores ocorreu, pela primeira vez, na construção do *Brown University Standard Corpus of Present-Day American English*, doravante *Brown Corpus*, o primeiro *corpus* computadorizado conhecido. O *Brown Corpus* foi criado por Kucera e Francis e, apesar do pequeno tamanho em relação aos padrões atuais (o *Brown Corpus* era composto por 1 milhão de palavras – um número condizente com as condições tecnológicas da época, em que a utilização de computadores restringia-se a poucos centros universitários e o armazenamento dos dados era feito em cartões perfurados), é considerado um marco no que se refere ao uso do potencial dos computadores para facilitar e racionalizar o registro, o armazenamento e a manipulação dos textos para elaboração de dicionários.

Embora o *Brown Corpus* tenha sido muito importante, a inovação nas práticas de produção de dicionários com o auxílio das tecnologias computacionais ganhou destaque na década de 1980, com o projeto *Collins Birmingham University*

International Language Database (COBUILD), da *University of Birmingham* em parceria com a Editora *Collins*, sob a coordenação do professor John Sinclair.

A partir da base de dados do COBUILD, foi construído o *Collins Cobuild English Language Dictionary*, o primeiro dicionário produzido por meio de um *corpus* eletrônico de língua, publicado em 1987. Segundo Rundell e Kilgarriff (2011), pela primeira vez, as entradas de um dicionário foram criadas e ilustradas a partir de um grande volume de dados reais de uso da língua inglesa. Os computadores foram utilizados desde a criação do COBUILD até a construção do *Collins Cobuild English Language Dictionary*, oferecendo o suporte tecnológico para a análise de textos e a aplicação de uma nova abordagem, baseada em *corpus*, em que cada fato linguístico identificado no *corpus* deveria se sustentar em evidências presentes em seus extratos.

No tocante às evidências, para Cabré (1999), lançar mão da abordagem baseada em *corpora* eletrônicos é significativamente vantajoso, em relação às maneiras anteriores de visualizar os dados linguísticos, ao propiciar aos terminógrafos e lexicógrafos a obtenção de informações empíricas sobre o uso real da língua e, por consequência, contribuir para a solidez no processo de elaboração de dicionários. Para compreendermos melhor essa afirmação, é interessante observarmos o que Tognini-Bonelli (2001) expõe sobre o assunto:

> A pesquisa com *corpus* pode ser vista como uma abordagem empírica em que, como nas demais investigações científicas, o ponto de partida são os dados autênticos. Portanto, o procedimento para a descrição de dados que faz uso de um *corpus* é indutivo, na medida em que os enunciados de natureza teórica sobre a língua ou a cultura são provenientes de observações de instâncias reais. A observação dos fatos da linguagem leva à formulação de uma hipótese para explicar esses fatos que, por sua vez, leva a uma generalização baseada na evidência da repetição de padrões de concordância. Por fim, surge um postulado teórico como resultado da unificação dessas observações (TOGNINI-BONELLI, 2001, p. 2)[31].

Conforme podemos notar no trecho citado, a observação dos fatos da língua, expressos em padrões recorrentes dos *corpora*, constitui-se como um elemento

[31] Esta e todas as traduções subsequentes são de nossa autoria. Original: "*Corpus work can be seen as an empirical approach in that, like all types of scientific enquiry, the starting point is actual authentic data. The procedure to describe the data that makes use of a corpus is therefore inductive in that it is statements of a theoretical nature about the language or the culture which are arrived at from observations of the actual instances. The observation of language facts leads to the formulation of a hypothesis to account for these facts; this in turn leads to a generalization based on the evidence of the repeated patterns in the concordance; the last step is the unification of these observations in a theoretical statement*".

importante para a criação de postulados teóricos sobre as instâncias de uso linguístico. Segundo Berber Sardinha (2004), os padrões aparecem por meio da "regularidade expressada na recorrência sistemática de unidades coocorrentes de várias ordens (lexical, gramatical, sintática etc.)" (BERBER SARDINHA, 2004, p. 40). Para a observação dos padrões, é necessário assumirmos a língua como "um sistema probabilístico de combinatórias, no qual uma unidade se define pelas associações que mantêm com outras unidades" (NOVODVORSKI; FINATTO, 2014, p. 15). Em especial, o uso do poder das ferramentas computacionais facilita a identificação de padrões associativos que possibilitam *insights* sobre o funcionamento da língua.

Nos anos que sucederam o projeto COBUILD, as tecnologias computacionais evoluíram rapidamente. Na década de 1990, os computadores diminuíram de tamanho e valor, deixaram de ser exclusividade de centros de pesquisa e universidades e passaram a fazer parte do cotidiano social com a criação do Computador Pessoal (*Personal Computer* – PC). Segundo Cabré (1999), nesse período, o uso dos PCs na prática de construção de dicionários tornou-se o padrão de trabalho. Os terminógrafos e lexicógrafos passaram a contar com um leque de ferramentas linguísticas baseadas em tecnologias computacionais para a manipulação, anotação, análise, conversão e armazenamento de dados. As tarefas repetitivas, antes realizadas manualmente, foram sendo resolvidas com o auxílio das máquinas. A necessidade da intervenção humana foi dispensada, em partes, em alguns casos, e, em outros, foi completamente substituída pelas novas ferramentas e recursos computacionais.

Rundell e Kilgarriff (2011) explicam que a transferência de tarefas repetitivas para os computadores trouxe benefícios em todos os sentidos, pois as máquinas são feitas com essa finalidade e as executam melhor do que os seres humanos. Os autores salientam que, ao serem poupados desse tipo de trabalho, os pesquisadores ficam livres para se dedicarem às demandas que exigem mais criatividade deles.

Se, por um lado, vários aspectos da produção de dicionários haviam sido facilitados, por outro lado, a coleta de textos ainda dependia da conversão deles para o formato eletrônico por meio de métodos manuais de digitalização. Por isso, a realização da coleta era considerada laboriosa e ainda representava um desafio para o desenvolvimento dos dicionários.

O progresso tecnológico nas áreas de *hardware* e *software*, instaurado nas décadas de 1980 e 1990, permitiu que a LC emergisse e se consolidasse. No entanto, o aparecimento dela, como a conhecemos hoje, só foi possível quando os computadores

foram conectados à Internet. A partir desse momento, conforme mencionam Rundell e Kilgarriff (2011), as pessoas ligadas à criação de dicionários passaram a contar com um dos mais importantes desenvolvimentos do século XXI, a Internet como *corpus* (*web as a corpus*).

Sob essa perspectiva, o problema da conversão dos textos para o formato eletrônico já não era tão grande, uma vez que, na Internet, eles já se apresentavam com tal configuração. Com a existência de produções escritas em formato digital e de novas tecnologias para a criação automática de *corpus* a partir dos materiais disponíveis na Internet, como o *WebBootCat* (BARONI *et al*., 2006), a coleta de grandes volumes de textos em curtos espaços de tempo tornou-se possível. As novas tecnologias de compilação automática de textos proporcionaram a redução do esforço e do tempo gastos para a elaboração de *corpora*, mas também suscitaram novos problemas para os linguistas – mais adiante tratamos disso de maneira aprofundada.

O uso das tecnologias computacionais para a produção de dicionários não está isento de embaraços. Cabré (1999) menciona que as principais dificuldades relacionadas à utilização dos recursos computacionais são a falta de integração com os métodos de trabalho, a incompatibilidade entre recursos computacionais, a constante necessidade de intervenção humana nas suas aplicações, a falta de interfaces de comunicação entre computadores e pessoas, a inadequação de equipamentos (*hardware*) e a indisponibilidade de *corpora* em línguas diferentes do idioma inglês.

Durante o período compreendido entre o levantamento dessas dificuldades e os dias atuais, os progressos científicos prosseguiram e as suplantaram em partes. Contudo, de acordo com Rundell e Kilgarriff (2011), ainda há espaço para melhorias nas tecnologias e, nesse ponto, nos interessa as melhorias relacionadas à utilização de tecnologias computacionais para a construção (compilação) manual de *corpora*. Na próxima seção, apresentamos os suportes teóricos e metodológicos da LC para a construção de *corpora*.

2.2 *Corpora*: conceitos e características

A Linguística é a área em que se desenvolve o estudo científico da linguagem humana com base em fatos linguísticos (MARTINET, 1978). De acordo com Widdowson (1996), de modo geral, os fatos linguísticos podem ser inferidos por meio da introspecção, da elicitação e da observação de dados provenientes do uso real da

língua pelos seus usuários. Widdowson (1996) esclarece que os fatos linguísticos apreendidos por meio da introspecção e da elicitação não revelam o uso efetivo da língua, pois partem das intuições que os seus usuários têm sobre ela. Já a observação de dados linguísticos decorrentes do uso real da língua e que refletem o comportamento linguístico de seus usuários constitui-se como uma forma mais segura para a realização de inferências sobre a língua. Nesse sentido, as análises linguísticas com base na LC podem ser consideradas altamente confiáveis, uma vez que partem da observação de *corpora* compostos por dados linguísticos reais.

Sinclair (2005) afirma que a construção de um *corpus* deve ser realizada de acordo com critérios bem definidos e eficientes o bastante para que o seu delineamento final possa garantir que o conjunto de textos seja representativo. O conceito de representatividade na LC está associado à capacidade que um *corpus* tem de representar uma língua ou uma variedade dela e ao modo como foi construído. Podemos dizer que um *corpus* é representativo quando, a partir da análise do conjunto de textos provenientes das várias situações comunicativas reais de uma comunidade linguística, é possível obter conclusões, a respeito de suas propriedades, que permitam generalizações sobre a língua ou sobre a variedade de língua em estudo.

A fase de construção de um *corpus* em que são definidos os seus critérios tem sido referenciada pelos autores da LC como o "desenho do *corpus*"[32]. Firmar o desenho de um *corpus* não é uma tarefa simples, pois, conforme Berber Sardinha (2004), não existem critérios objetivos para isso. Segundo Blecha (2012), a delimitação do desenho de um *corpus* deve ser orientada em consonância com os objetivos da pesquisa. Tagnin (2010) coaduna com Blecha (2012) e afirma que cabe ao criador do *corpus* a responsabilidade de definir os critérios que possam garantir sua representatividade.

Com base no posicionamento desses autores, podemos dizer que o pesquisador, tendo em mente os objetivos da pesquisa, precisa estabelecer as variáveis quantitativas e qualitativas que, de fato, tornarão o *corpus* representativo. A fim de compreendermos melhor a importância dos critérios do desenho de um *corpus* para sua representatividade, tomamos, como exemplo, os fundamentos e implicações referentes à

[32] Na literatura da LC, em língua inglesa, encontramos o termo *corpus design*.

extensão, ao tipo e à estrutura conceitual[33] de um *corpus* adotados para a organização de seus dados linguísticos.

A extensão ou o tamanho de um *corpus* representa o volume de dados linguísticos disponíveis para análise. Na literatura da LC, não encontramos a definição exata do tamanho necessário para que um *corpus* seja representativo. No entanto, para estudos que consideram a chavicidade[34] de palavras, encontramos recomendações e estimativas, como a de Berber Sardinha (2004), que afirma que a relação de tamanho entre os *corpora* de estudo e os *corpora* de referência influencia a quantidade de palavras-chave obtidas. Para fins práticos de pesquisa, o conhecimento dessa relação nem sempre serve como diretriz para a definição do tamanho do *corpus*, como aponta Berber Sardinha (1999):

> Sabendo-se a influência de um *corpus* de uma certa dimensão na chavicidade das palavras, é possível planejar qual o tamanho ideal dos *corpora*. E tendo-se conhecimento do tamanho ideal dos *corpora*, torna-se possível planejar a pesquisa de modo que não se desperdice recursos coletando-se dados além do que seria teoricamente necessário. O pesquisador poderia, então, saber qual o impacto que um *corpus* de tamanho x teria nos resultados de sua pesquisa, e planejar sua coleta de dados conscientemente (BERBER SARDINHA, 1999, p. 4).

Além de atentar para a relação de tamanho entre os *corpora* da pesquisa, Berber Sardinha (2004) expõe que um *corpus* deve ser o mais extenso possível:

> O *corpus* é uma amostra de uma população cuja dimensão não se conhece (a linguagem como um todo). Desse modo, não se pode estabelecer qual seria o tamanho ideal da amostra para que represente essa população. Uma salvaguarda é tornar a amostra a maior possível, a fim de que ela se aproxime ao máximo da população da qual deriva, sendo, portanto, mais representativa (BERBER SARDINHA, 2004, p. 23).

Mesmo com a recomendação de Berber Sardinha (2004), o aumento constante do volume de informações disponíveis na Internet e os progressos tecnológicos para o seu processamento, cabe pontuar que a extensão de um *corpus* está sujeita à disponibilidade de dados que atendam às especificidades do desenho dele. A obtenção

[33] Considerando que a nossa pesquisa envolve a aplicação do *ToGatherUp* na construção do protótipo de um *corpus* para fins de estudo terminológico, focamos a nossa atenção na estrutura conceitual tipicamente usada em pesquisas terminológicas – Árvore de Domínio.

[34] De acordo com Fromm (2007), a chavicidade (*keyness*) informa o quanto uma palavra se destaca na relação entre a sua frequência no *corpus* de estudo e no *corpus* de referência.

de dados suficientes para cada campo semântico de uma Árvore de Domínio, no caso das pesquisas terminológicas, ou para cada gênero textual que compõe um *corpus* de estudo do léxico, de modo que seja garantido o balanceamento[35] do *corpus*, é um exemplo dessa situação.

Ademais, Fromm (2003) chama a atenção para o fato de que o desenvolvimento de um *corpus* extenso requer a participação de vários pesquisadores e auxiliares; caso contrário, a construção dele pode demorar anos para ser concluída. Nessa situação, há a questão do tempo que o pesquisador (ou a equipe de pesquisadores) tem para dedicar à obtenção de dados.

Berber Sardinha (1999) observa que, na prática, "o pesquisador coleta uma certa quantidade de dados de acordo com suas possibilidades, efetua a análise, mas não sabe se sua coleta foi além ou aquém do que seria teoricamente mais adequado" (BERBER SARDINHA, 1999, p. 4). Por essa razão, Nelson (2010) afirma que a criação de um *corpus* é "uma aceitação entre o que é o esperado e o que é possível" (NELSON, 2010, p. 30)[36] e Meyer (2004) explica que as mudanças no desenho inicial do *corpus* são naturais e inevitáveis (desde que não comprometam a integridade do *corpus*) diante dos obstáculos e complicações que podem surgir durante a sua compilação.

Em relação à definição do tipo de *corpus*, a elaboração de dicionários envolve o uso de *corpora* gerais ou especializados de língua em pesquisas lexicográficas e a utilização de ambos em pesquisas terminográficas. Os *corpora* gerais são grandes volumes de textos provenientes das várias possibilidades de instanciação de uma língua, distribuídos de forma homogênea, coletados com o objetivo de representá-la como um todo. A representatividade dos *corpora* gerais faz com que eles sirvam como base para a investigação do vocabulário e dos padrões linguísticos de uma língua, sendo indispensáveis em pesquisas lexicográficas que visem à produção de dicionários gerais de língua. Além disso, os *corpora* gerais podem ser utilizados em pesquisas terminográficas, nas quais os pesquisadores precisam constituir *corpora* especializados, mas também necessitam de *corpora* de referência para obterem a lista de palavras-chave (*keywords*); nesses casos, os *corpora* gerais são usados como *corpora* de referência, uma vez que estes últimos, de acordo com Berber Sardinha (2004), devem ser cinco

[35] Aluísio e Almeida (2006) definem o balanceamento como o equilíbrio entre as categorias atribuídas aos textos que compõem um *corpus*. Os gêneros discursivos, os tipos de textos e os autores são exemplos dessas categorias.

[36] Original: "*any attempt at corpus creation is therefore a compromise between the hoped for and the achievable*".

vezes maiores que o tamanho do *corpus* de estudo para que possam retornar "significativamente mais palavras-chave do que *corpora* de tamanhos menores" (BERBER SARDINHA, 2004, p. 102).

O *British National Corpus* (BNC), por exemplo, é um dos principais *corpora* gerais da língua inglesa, contendo 100 milhões de palavras de falantes nativos do inglês britânico, presentes em uma gama de textos compilados com o propósito de representar tal variedade. Na língua portuguesa, temos o *Corpus* Brasileiro[37], um *corpus* geral com aproximadamente um bilhão de palavras do português brasileiro, criado na Pontifícia Universidade Católica (PUC) de São Paulo sob a coordenação de Tony Berber Sardinha.

Consoante Sinclair (1996), a função de *corpora* gerais é "fornecer informações abrangentes sobre o idioma" (SINCLAIR, 1996, n. p.)[38]. Sob essa ótica, os *corpora* gerais tendem a ser grandes o bastante para que representem "todas as variedades relevantes da língua e seu vocabulário de modo que possa ser usado como base para gramáticas, dicionários, tesauros e outros materiais de referência" (SINCLAIR, 1996, n. p.)[39]. Reppen (2010) reforça essa ideia ao explanar que *corpora* grandes parecem assegurar melhor a possibilidade de obtenção dos sentidos de uma palavra nas situações em que "todos os sentidos de uma palavra precisam ser capturados" (REPPEN, 2010, p. 32)[40].

Se, por um lado, os *corpora* gerais são essenciais para a descrição de uma língua, por outro, "eles são menos condutivos para a análise do uso da língua em situações acadêmicas e profissionais específicas" (CONNOR; UPTON, 2004, p. 2). Portanto, nesses contextos, surge a necessidade de o pesquisador usar *corpora* especializados, que apresentam uma amostra mais focada de determinados aspectos linguísticos a serem observados por ele. Nessa perspectiva, Kübler e Aston (2010, p. 507) explicam que os *corpora* especializados são mais propensos a documentar as convenções de gênero e os conceitos e termos de um domínio.

Connor e Upton (2004) mencionam que:

> Ao invés de serem compilados para representar a linguagem em seus vários propósitos comunicativos, os *corpora* especializados,

[37] Disponível em: http://corpusbrasileiro.pucsp.br. Acesso em: 21 jun. 2018.
[38] Original: "*to provide comprehensive information about language*".
[39] Original: "*all the relevant varieties of the language, and the characteristic vocabulary, so that it can be used as a basis for reliable grammars, dictionaries, thesauruses and other language reference materials*".
[40] Original: "*all senses of a word need to be captured*".

> geralmente, focam em um gênero específico (por exemplo: documentos de pesquisa e cartas comerciais) ou em uma situação específica (por exemplo: palestras acadêmicas e comunicação empresarial) (CONNOR; UPTON, 2004, p. 2)[41].

Acrescentamos que os *corpora* especializados podem, ainda, se limitar a textos característicos de uma comunidade, de uma área de conhecimento ou de um tipo específico de língua, como as línguas de especialidade. Assim, esclarecemos que *corpora* especializados podem ser utilizados tanto em pesquisas lexicográficas, em específico, naquelas em que se pensa na produção de um dicionário que se restrinja a uma questão particular de uma língua, como o trabalho de Grama (2016), que elaborou fichas lexicográficas para os elementos coesivos sequenciais da língua portuguesa com base em um *corpus* de redações, quanto para estudos terminográficos que visem à construção de um dicionário com termos de uma área do conhecimento, por exemplo, a pesquisa de Cardoso (2017), que produziu o TermosTeo – um vocabulário da área da Teologia, conduzido por *corpora*.

Em relação à extensão, os *corpora* especializados tendem a ser menores que os gerais. Koester (2010) justifica que os especializados não precisam ter a extensão dos *corpora* gerais da seguinte maneira:

> A razão para isso é que, como os *corpora* especializados são cuidadosamente direcionados, eles são mais propensos a representar adequadamente um tipo de registro ou de gênero do que *corpora* gerais. Mesmo quando possuem quantidades relativamente pequenas de dados, o léxico e as estruturas especializadas ocorrem com mais regularidade e distribuição de padrões do que em um *corpus* geral (KOESTER, 2010, p. 68-69)[42].

Além da extensão e do tipo de *corpus*, para Fromm (2013), quando pretendemos trabalhar com a descrição terminológica de alguma área de especialidade, é essencial que seja feita a elaboração de sua Árvore de Domínio, pois é a partir dela que se estabelece o planejamento e o balanceamento de um *corpus*. Segundo o autor, a

---

[41] Original: "*Instead of being compiled for the representativeness of language across a large number of communicative purposes, specialized corpora often focus on one particular genre (e.g. research papers, letters of business requests) or a specific situation (e.g. academic lectures, office communication in business)*".

[42] Original: "*The reason for this is that as specialised corpora are carefully targeted, they are more likely to reliably represent a particular register or genre than general corpora. Even with relatively small amounts of data, 'specialized lexis and structures are likely to occur with more regular patterning and distribution' than in a large, general corpus*".

coleta das informações para a composição de um banco de dados de acordo com uma Árvore de Domínio promove a organização das informações.

O termo Árvore de Domínio é usado para referenciar o sistema de classificação e de representação epistemológica de um campo do conhecimento, envolvendo as inter-relações entre áreas e subáreas. A Árvore de Domínio também pode ser chamada de Árvore do Conhecimento, Árvore de Especialidade, Conceitualização ou Taxonomia de uma área. Krieger e Finatto (2004, p. 134) definem a Árvore de Domínio como um "diagrama hierárquico composto por termos-chaves de uma especialidade, semelhante a um organograma", que permite uma visão geral e ampla do objeto de estudo de uma especialidade, auxilia na compreensão das hierarquias básicas e situa o recorte terminológico do projeto em desenvolvimento.

Conforme Cabré (2003), a alocação das unidades terminológicas de um *corpus* em lugares precisos determina o significado específico delas e deixa suas relações semântico-conceituais evidentes. Por essa razão, Souza (2011, p. 48) afirma que a definição dos termos "deve ser realizada dentro do campo nocional ou conceitual" correspondente ao discurso de especialidade a que eles pertencem. Bononno (2000, p. 651) esclarece que a organização de dicionários em ordem alfabética não é suficiente para que sejam estabelecidas as relações semântico-conceituais dos termos por não oferecer a "indicação de arranjo sistemático e, consequentemente, nenhuma informação sobre as relações entre conceitos de uma determinada área".

A definição da Árvore de Domínio de um projeto terminográfico pode se dar pela adoção de uma estrutura conceitual já existente ou pela criação de outra. Ao optar pela elaboração de uma nova estrutura conceitual, o pesquisador deverá estar ciente de que ela poderá ser alvo de críticas sobre a representatividade dela. Isso ocorre porque a organização do conhecimento (e da comunicação) de uma comunidade discursiva está sujeita aos diferentes pontos de vistas e necessidades dos indivíduos que a constituem. Em outras palavras, a falta de consenso entre os membros de uma comunidade dificulta o estabelecimento da Árvore de Domínio padrão para determinada área do conhecimento. A consequência disso é a coexistência de diversos tipos de classificação formados, de acordo com Barbosa (1995), a partir de diferentes recortes da realidade.

2.3 A construção de um *corpus*

Os princípios da LC não tratam de um modelo sistematizado para a construção manual de um *corpus*. O que encontramos na literatura são abordagens que, embora obedeçam às diretrizes criadas por Sinclair (2005), diferenciam-se entre si em aspectos de organização, uso de ferramentas e técnicas. Ou seja, as pesquisas não seguem um padrão no que diz respeito ao modo de criar um *corpus*.

Essa ausência de um modelo a ser seguido na montagem de um *corpus* pode ser justificada pela combinação entre a gama de recursos que a LC oferece, a diversidade de campos em que a metodologia da LC pode ser aplicada e os variados objetivos de pesquisas. Em conjunto, isso constitui um ambiente propício à adoção de diferentes estratégias de trabalho pelos pesquisadores no desenvolvimento de suas investigações científicas. Diante disso, podemos afirmar que a aplicação de um modo fixo e ideal de construir um *corpus* é, praticamente, inviável.

Cabe destacar que, ao mesmo tempo em que a inexistência de um padrão promove a flexibilidade das práticas de elaboração de *corpora*, ela também gera problemas como a variação significativa dos nomes que são atribuídos às ações que envolvem a construção de um *corpus*. Nesse sentido, há autores que se referem ao trabalho de criação de *corpus* como um processo dividido em estágios (cf. ATKINS; CLEAR; OSTLER, 1992; ESCARTIN, 2012; KENNEDY, 1998), em ciclos (cf. BIBER, 1993) ou em passos (cf. SANTOS, 2011). Além dessas denominações, é comum encontrarmos palavras, tais como: “tarefas”, “atividades” e “procedimentos”, sendo utilizadas com o mesmo sentido, isto é, remetendo-se às mesmas ações.

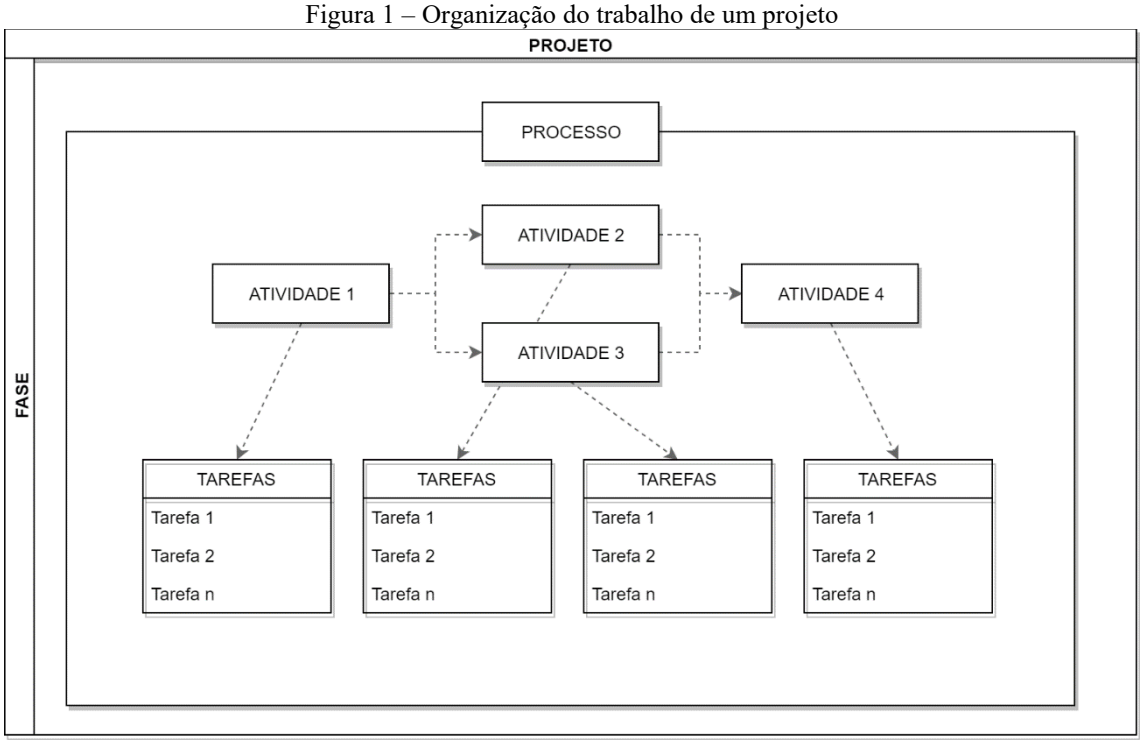
Para não repetirmos essa prática, decidimos adotar uma nomenclatura e aplicar conceitos da área de Gerenciamento de Projetos ao longo da nossa pesquisa, por considerarmos que a construção de um *corpus* enquadra-se no conceito de projeto e, em partes, nos princípios dessa área. De acordo com o guia *Project Management Body of Knowledge* (PMBOK), publicado em 2013 e considerado como a principal referência da área de Gerenciamento de Projetos, um projeto é “um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo” que possui um “ciclo de vida” (PMBOK, 2013).

O ciclo de vida de um projeto corresponde à sequência de fases pelas quais ele passa ao longo do seu desenvolvimento. Do mesmo modo que a gama de recursos da LC, a natureza e os objetivos de uma pesquisa delineiam a forma como um *corpus* é

construído, as fases do ciclo de vida de um projeto são estabelecidas pelas necessidades de gerenciamento e controle do projeto, pela sua natureza e sua área de aplicação.

Durante o ciclo de vida do projeto, cada fase pode comportar um ou mais processos. Estes, por sua vez, podem admitir uma ou mais atividades. Uma atividade pode relacionar-se com outra(s), de maneira lógica, de modo que seu início ou sua continuidade somente seja possível após a geração de um ou mais resultados (entregas) de outra(s) atividade(s). No nível mais baixo do ciclo de vida de um projeto, encontram-se as tarefas, que são as menores unidades de trabalho possíveis pertencentes ao escopo de uma atividade. A Figura 1 ilustra as relações dos componentes do trabalho de um projeto apresentadas neste parágrafo.

Figura 1 – Organização do trabalho de um projeto



Fonte: o autor.

A nomenclatura e a organização mostradas na Figura 1 foram utilizadas como parâmetros para a redação desta pesquisa. Por essa razão, consideramos que a construção manual de um *corpus* é equivalente à realização de um projeto, composto por fases, processos, atividades e tarefas. Na sequência, apresentamos as contribuições teóricas dos autores da LC, tentando situá-las, de forma mais adequada possível, à organização de um projeto.

2.4 O projeto de construção de *corpus*

O projeto de construção manual de um *corpus*, de modo geral, pode ser dividido em três fases distintas: a) a inicial, em que há o planejamento do *corpus*; b) a intermediária, caracterizada pela obtenção, preparação e armazenamento dos dados do *corpus* e c) a de encerramento, na qual ocorre a distribuição dos dados do *corpus*. A seguir, discorremos, detalhadamente, sobre cada etapa.

2.4.1 Fase inicial do projeto

A fase inicial do projeto de construção manual de um *corpus* é caracterizada pela execução das atividades de planejamento do *corpus*, de definição dos recursos necessários para elaborá-lo e de esquematização do cronograma de execução do projeto. No próximo tópico, discorremos sobre elas.

2.4.1.1 O processo de planejamento do corpus

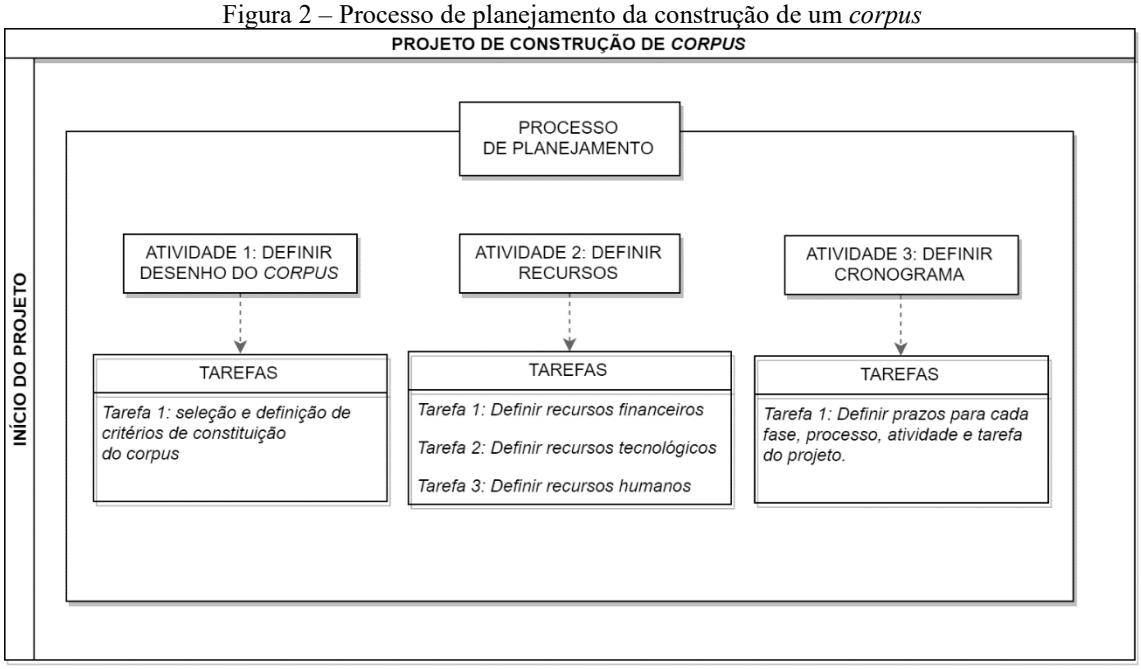
O êxito na realização de um projeto está diretamente relacionado às decisões que são tomadas durante o planejamento do *corpus*. De acordo com Nelson (2010, p. 53), há uma série de variáveis que precisam ser consideradas antes do início da compilação dos dados, a saber: o tamanho do *corpus*, o balanceamento dele, a estrutura conceitual em que os textos serão organizados, o formato de armazenamento dos textos, a maneira como será feita a coleta dos textos, o padrão que será usado para a nomeação dos arquivos e o controle em relação à coleta e ao gerenciamento dos textos.

Algumas dessas questões são analisadas durante o desenho do *corpus*, que é a primeira atividade do processo de planejamento do *corpus*. Para o estabelecimento do desenho do *corpus*, o seu criador precisa executar as tarefas de seleção e definição dos critérios que nortearão a constituição do *corpus*.

Ademais, Atkins, Clear e Ostler (1992, p. 3) mencionam o fato de que o planejamento do *corpus* deve prever o uso de recursos financeiros, tecnológicos e humanos necessários para garantir a conclusão do projeto. Santos (2011) complementa as exigências do planejamento do *corpus* ao afirmar que é necessário estabelecer o cronograma para a execução do projeto, pois várias decisões que precisam ser tomadas durante a elaboração do *corpus* estão vinculadas às restrições de tempo para a sua

realização. A Figura 2 ilustra o processo de planejamento e as atividades da fase inicial do projeto de construção de um *corpus*.

Figura 2 – Processo de planejamento da construção de um *corpus*



Fonte: o autor.

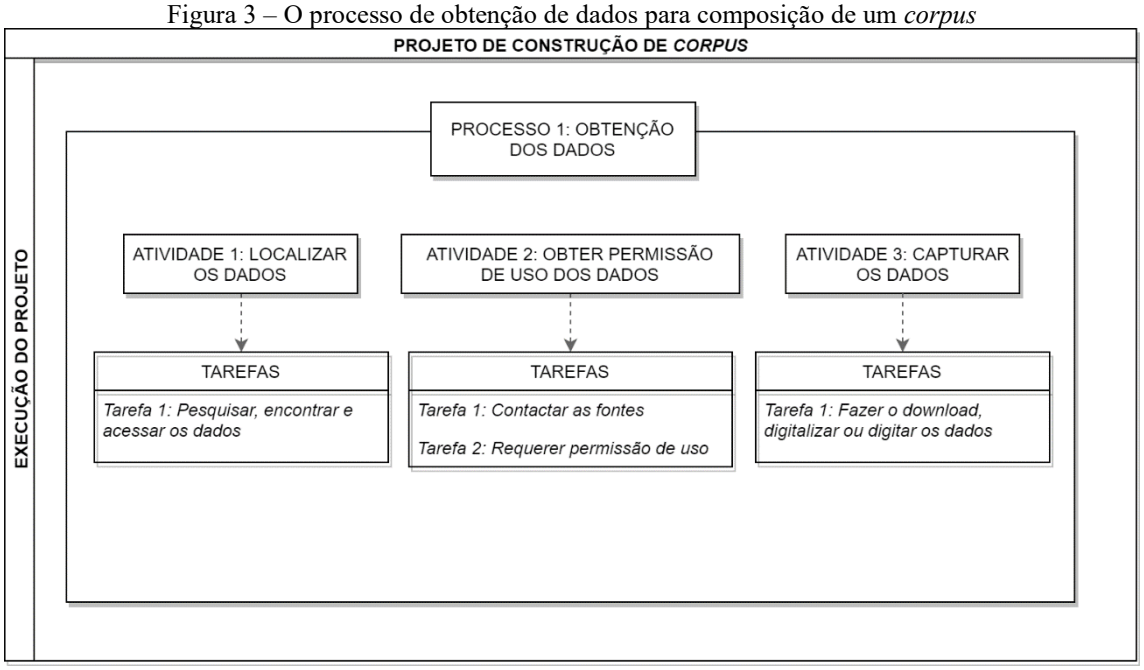
2.4.2 A fase de execução do projeto

A fase de execução sucede a fase inicial do projeto e caracteriza-se pela realização das atividades relativas aos processos de obtenção, preparação e armazenamento de dados do *corpus*. Tais processos são descritos a seguir.

2.4.2.1 O processo de obtenção dos dados

Após planejar a construção do *corpus*, o pesquisador tem em mãos os parâmetros que o guiarão no processo de obtenção de dados linguísticos com o propósito de compor o *corpus*, conforme mostra a Figura 3.

Figura 3 – O processo de obtenção de dados para composição de um *corpus*



Fonte: o autor.

A primeira atividade desse processo consiste na pesquisa, localização e acesso aos materiais que comportam os dados desejados. Para Sinclair (1991), os dados podem ser encontrados, em suas versões originais, na forma eletrônica, impressa ou escrita à mão[43]. Esse autor salienta que a obtenção de textos no formato eletrônico é a mais fácil e desejável comparada às demais, visto que exige um menor esforço do pesquisador no momento de adaptá-los para posterior processamento feito pelas ferramentas computacionais.

A principal fonte de dados em formato eletrônico da atualidade é a Internet. Nela, os textos podem estar sob a forma de documentos de hipertexto[44], de arquivos eletrônicos de diferentes formatos – *Microsoft Word document* ou *Word Binary File Format* (DOC ou DOCX), em *Hypertext Markup Language* (HTML), em

[43] A afirmação de Sinclair (1991) refere-se somente aos dados de produção escrita e não considera dados provenientes da produção oral. Assim como a afirmação de Sinclair (1991), a proposta de organização de projetos de construção de *corpora* desta pesquisa, também, delimitou-se aos dados de produção escrita, uma vez que os textos que compuseram o CoCLI foram, exclusivamente, provenientes desse tipo de produção. Apesar desse fato, temos consciência de que os dados de *corpora* podem ser provenientes tanto da modalidade escrita quanto da modalidade oral da língua que, conforme observam Mello e Raso (2014), requerem mais esforço para a sua compilação. Um aspecto comum dos dois tipos de produção é que os dados são produzidos por seres humanos. Acreditamos que esse aspecto sofrerá mudanças, em um futuro breve, com a inclusão de textos produzidos por inteligência artificial como, por exemplo, o texto criado pelo modelo GPT-2 (RADFORD *et al.*, 2018), da empresa OpenAi[1] (Disponível em https://openai.com/blog/better-language-models/. Acesso em: 15 abr. 2019), presente no **ANEXO B – Exemplo de texto produzido pelo modelo GPT-2** deste trabalho.

[44] De acordo com Baker, Hardie e Mcenery (2006), um documento de hipertexto pode conter *links* para outros documentos e formar redes de textos. Os documentos de hipertexto estão presentes na Internet sob o formato HTML, uma derivação do *Standard Generalised Markup Language* (SGML).

OpenDocument (ODT), em *Portable Document Format* (PDF), em *Rich Text Format* (RTF), em *Extensible Markup Language* (XML) e em *Plain Text Format* (TXT) – e, ainda, como resultado de consultas a plataformas (por exemplo: *Twitter* e *Facebook*) que oferecem interfaces, conhecidas como *Application Programming Interface* (API)[45], que permitem a interação entre diferentes tipos de aplicações e suas bases de dados.

Nas situações em que os textos estão no formato eletrônico, a coleta pode ser efetuada pelo próprio esforço do pesquisador ou automaticamente pelas ferramentas de compilação de *corpus* mencionadas anteriormente. A captura manual de textos em HTML pode ser feita através da transferência dos seus dados para um dos diversos tipos de arquivos digitais voltados para textos por meio de uma sequência apropriada de comandos de seleção[46], cópia[47] e colagem[48] de caracteres de textos ou pelo salvamento da página em que o texto se apresenta sob um dos formatos de arquivos de texto. Já a captura manual dos textos disponíveis sob o formato de arquivos eletrônicos, como o PDF, precisa ser realizada de forma manual pelo pesquisador por meio de *download*, de acordo com as funcionalidades de *download* de arquivos oferecidas pelos navegadores de Internet.

No que diz respeito aos textos impressos, estes precisam ser convertidos para o formato eletrônico por meio de digitalização ou digitação. O método preferível de conversão é a digitalização com o uso de equipamentos digitalizadores, conhecidos como *scanners*, em combinação com *softwares* OCR. Nelson (2010) recomenda que os textos digitalizados sejam checados cuidadosamente pelo pesquisador com a finalidade de certificar-se de que a versão digital corresponde à versão original e alerta que essa é uma tarefa demorada e que não pode ser dispensada.

Outra maneira de digitalizar textos impressos é a digitação. Segundo Kennedy (1998), esse é o caminho mais básico e o mais demorado para a captura de um texto e somente é adotado quando outros métodos não são possíveis. Kennedy (1998) considera a digitação aplicável quando a digitalização do texto com *scanners* não é possível,

---

[45] Uma API é uma interface computacional que permite que duas aplicações possam conversar entre si.

[46] A seleção de textos disponíveis em páginas de Internet pode ser feita através de comandos de seleção que variam de acordo com a necessidade do usuário e com o navegador em que a página for aberta. No navegador *Google Chrome*, por exemplo, o comando CTRL+A (Comand + A, no MAC) faz com que todos os textos disponíveis em uma página sejam selecionados.

[47] A maior parte dos navegadores de Internet apresenta o comando CTRL+C (Comand + C, no MAC) para a cópia de textos.

[48] A maior parte dos programas de texto apresenta o comando CTRL+V (Comand + V, no MAC) para a colagem de textos armazenados na memória do computador.

devido à má qualidade do material original ou à presença de caracteres de difícil reconhecimento pelos *softwares* OCR.

No caso de textos escritos à mão, a digitação é obrigatória. Nelson (2010) levanta, ainda, a possibilidade do uso combinado de métodos para a captura de determinado texto. Vale ressaltar que, independentemente do modo escolhido para obter os textos de um *corpus*, o pesquisador deverá observar os aspectos éticos e legais referentes ao uso deles.

Para Mcenery e Hardie (2011), a inobservância dos direitos autorais (*copyright*) de um texto pode comprometer a legalidade de um *corpus*. Por essa razão, os autores recomendam que o pesquisador entre em contato com os detentores dos direitos de propriedade intelectual do texto para obter permissão de uso antes de incorporá-lo em um *corpus*. Na mesma direção, Nelson (2010) lembra que o pesquisador deve ficar atento às diferenças entre as leis de direitos autorais de cada país.

No Brasil, por exemplo, o direito autoral é regulado pela Lei de Direitos Autorais[49] (BRASIL, 1998) e, de acordo com ela, a reprodução e a distribuição não autorizada de obras protegidas por direitos autorais configuram-se como violação. Portanto, a inclusão de um texto protegido pela referida lei em um *corpus* somente é possível após a autorização dos detentores dos direitos autorais desse texto, principalmente, quando há a intenção, por parte do pesquisador, de disponibilizá-lo para o público.

Com o fito de contornar os possíveis problemas decorrentes de restrições de uso de textos, Nelson (2010) sugere optar por aqueles que estão livres e disponíveis em bases como o Projeto Gutenberg[50]. Mcenery e Hardie (2011) explanam sobre outro modo de resolver a questão das restrições de direitos: a utilização dos textos sem qualquer preocupação quanto à obtenção da permissão de uso desde que não ocorra a distribuição deles.

Na abordagem proposta por Mcenery e Hardie (2011), os dados do texto são apresentados no escopo de ferramentas computacionais, como os concordanciadores[51], sob a forma de pequenos recortes que não chegam a infringir os direitos autorais do texto. Nesse caso, os autores afirmam que o pesquisador pode indicar os endereços das

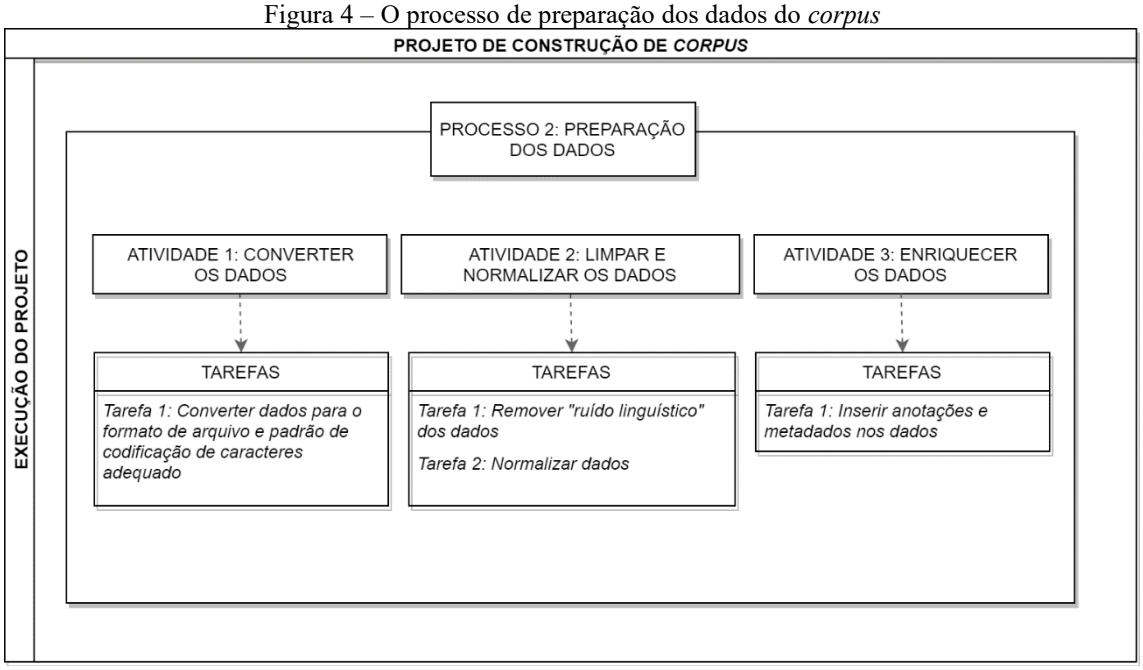
[49] Disponível em: http://www.planalto.gov.br/ccivil_03/LEIS/L9610.htm. Acesso em: 18 dez. 2018.
[50] O Projeto Gutemberg (www.gutenberg.org) oferece milhares de livros eletrônicos, cujos direitos autorais foram expirados e passaram a pertencer ao domínio público.
[51] De acordo com Tagnin (2010), um concordanciador é um "programa que extrai todas as ocorrências de uma palavra de busca num *corpus* juntamente com seu contexto, apresentando-as na forma de uma concordância" (TAGNIN, 2010, p. 358).

fontes em que os textos foram encontrados ao invés de compartilhar os arquivos do *corpus* com outras pessoas. Para eles, essa seria uma forma de permitir que outros pesquisadores possam ter acesso aos dados de um *corpus* para fins de verificação e replicação de análises sem que exista qualquer possibilidade de infração de direitos autorais. Cabe lembrar que essa última abordagem não é uma solução ideal, uma vez que depende da permanência dos endereços dos textos nas fontes utilizadas na pesquisa.

2.4.2.2 O processo de preparação dos dados do corpus

Após a obtenção dos textos, o pesquisador precisa certificar-se de que eles são "úteis" para a inclusão em um *corpus*. A utilidade de um texto, para as pesquisas da LC, está associada, obrigatoriamente, à condição favorável dele para o processamento através de ferramentas computacionais e, opcionalmente, à integridade e ao enriquecimento dele. Esses aspectos que configuram a preparação dos dados do *corpus* são contemplados na Figura 4.

Figura 4 – O processo de preparação dos dados do *corpus*



Fonte: o autor.

O processamento de um texto por meio de uma ferramenta computacional exige que ele esteja em um formato "compreensível pelos computadores" (*machine-readable*). No entanto, segundo Shiver (1997 *apud* BATEMAN, 2008), grande parte do conhecimento disponível na sociedade manifesta-se sob a forma de textos escritos

engendrados de modo que a interpretação deles seja a mais conveniente possível para os seres humanos[52].

Shiver (1997 *apud* BATEMAN, 2008) afirma que a produção de textos envolve diferentes formas de representação das informações, como o uso de elementos visuais (imagens, tabelas, gráficos e vídeos), de conteúdo (*hiperlinks*) e de estrutura (*layout*), no intuito de alcançar um fim comunicativo com seu público-alvo. Porém, os recursos que tornam um texto propício à interpretação humana podem ser prejudiciais ao processamento feito pelos computadores, já que estes ainda não possuem as mesmas capacidades de decodificação que os homens. Em virtude disso, na metodologia da LC, é indispensável a conversão das versões originais de textos, eletrônicos ou não, que fazem parte do *corpus* para formatos e codificações de caracteres apropriados ao trabalho que a máquina executará.

A codificação de caracteres de um texto é um dos aspectos importantes para que ele seja processado de forma adequada por um computador. Apesar de ser um assunto bastante técnico, acreditamos que é interesse da comunidade linguística a compreensão de termos, tais quais: *American Standard Code for Information Interchange* (ASCII) e *Unicode UTF-8*, usados em frases como "um arquivo de texto puro em formato ASCII".

De acordo com Leggett (2014), o ASCII foi um dos primeiros padrões estabelecidos para a associação de sequências binárias de dígitos (zeros e uns) a caracteres. A autora explica que, ao pressionarmos uma tecla no computador, é enviado um sinal elétrico que é decodificado sob a forma de uma sucessão de números (zeros e uns). Cada um dos caracteres de uma língua é representado por uma sequência específica de números definida pelo padrão de codificação.

A primeira versão do ASCII era capaz de associar apenas 158 caracteres às sequências em questão, o suficiente para representar todas as letras do alfabeto da língua inglesa (maiúsculas e minúsculas) e das línguas que compartilhavam esse alfabeto, os números e os caracteres básicos de pontuação. A segunda versão do ASCII (Latin-1) expandiu a sua capacidade para 256 caracteres e possibilitou a inclusão deles acompanhados de sinais diacríticos. Todavia, isso não foi suficiente diante de todos os caracteres das línguas existentes.

[52] Dietrich *et al*. (2009) definem *human-readable* como os textos que possuem um formato orientado para o consumo dos seres humanos.

A solução para esse problema foi a criação do *Unicode standard*[53], um padrão desenvolvido com o objetivo de representar todos os caracteres existentes possíveis a sequências numéricas binárias do computador. O *Unicode*, de acordo com o documento "*UTF-8*, *a transformation format of ISO 10646*"[54], do *RCF Editor*[55], apresenta a forma *UTF-8* como padrão. Desse modo, o *Unicode UTF-8* possui a vantagem de ser compatível com todas as línguas, sendo, então, recomendado para a construção de *corpus* por autores como Anthony (2018), Schäfer e Bildhauer (2013). A nosso ver, essa orientação parece-nos sensata principalmente diante do alerta feito por Santos (2011) sobre a possibilidade de impactos negativos na análise de textos caso a codificação dos arquivos em que foram inseridos seja incompatível com a língua em que os textos foram escritos.

O formato do arquivo em que o texto é salvo também é outro aspecto importante para o processamento. As informações textuais podem ser arquivadas em DOCX, HTML, ODT, PDF, RTF, XML e em TXT. Podemos dividi-los em três grupos: a) formatos orientados para *layout* de páginas (DOCX, ODT, RTF e PDF); b) formatos orientados para anotação (*markup*) estrutural (HTML) ou de conteúdo (XML) e c) formatos orientados somente para texto (TXT).

Os formatos orientados para *layout* de páginas e o HTML são os mais utilizados para a divulgação de textos e possuem diferentes modos de apresentação de informações. Os que melhor se adequam ao processamento dos computadores são os orientados para anotação e somente para texto. Entre esses últimos, o TXT é o que possui maior relevância para o arquivamento de textos de um *corpus*, pois lida com textos puros (*plain texts ou raw data*)[56] – constituídos apenas de letras, números, símbolos e espaços e isentos de informações sobre formatação.

Assim, podemos dizer que o TXT torna-se conveniente para o processamento dos computadores, uma vez que o uso dele vai ao encontro de pesquisas linguísticas que têm interesse em analisar somente dados linguísticos. Indubitavelmente, é um formato amplamente utilizado para o arquivamento de textos.

---

[53] O padrão foi criado pelo *Unicode Consortium* (www.unicode.org).
[54] Disponível em: https://www.rfc-editor.org/info/rfc3629. Acesso em: 22 dez. 2018.
[55] O *RFC Editor* foi fundado em 1998 pela *Defense Advanced Research Projects Agency* (DARPA) do governo dos Estados Unidos da América para a realização da edição, publicação e catalogação das RFCs, as séries de documentos com as informações técnicas e organizacionais sobre a Internet. Disponível em: https://www.rfc-editor.org/. Acesso em: 6 mar. 2019.
[56] De acordo com Baker, Hardie e Mcenery (2006), *plain text* é um texto que contém somente palavras de um documento original e que não apresenta qualquer tipo de etiquetagem.

Vale enfatizar que a conversão dos textos que compõem um *corpus* para o TXT, com a devida codificação, faz com que eles percam todas as formatações presentes nos arquivos em que estavam na versão original (BOWKER; PEARSON, 2002), mas, segundo Santos (2011), os arquivos convertidos podem apresentar resíduos como, por exemplo, números de páginas, informações de cabeçalhos e rodapés de páginas, anotações sobre a divisão das seções do texto, conteúdos de tabelas (que perdem o sentido ao serem desprovidos da estrutura da tabela) e erros de codificação de caracteres (resultantes da transposição de um padrão de codificação para outro).

Os resíduos nos arquivos TXT são vistos por Edward (2015) como dados "sujos". Nas subáreas da Computação, eles são chamados de "ruído" e caracterizam-se pela possibilidade de não serem compreendidos ou de serem interpretados incorretamente pelos computadores. No contexto dos dados linguísticos, os resíduos presentes nos arquivos TXT podem ser considerados como ruído linguístico e podem gerar problemas no que diz respeito aos métodos da LC (Edward, 2015), que se baseia na análise da quantidade de vezes que determinado elemento linguístico apareceu em um *corpus*.

Conforme Gries (2009), a frequência de um elemento linguístico, isto é, o resultado obtido ao somar o número de vezes em que ele ocorreu num *corpus*, é base para a formação de listas de palavras. Essas são utilizadas para a construção de listas de palavras-chave que, de acordo com Tagnin (2015) e Edward (2015), apresentam os elementos linguísticos cujas frequências são estatisticamente relevantes a partir do resultado da comparação entre listas de palavras de um *corpus* de estudo e um *corpus* de referência.

Pensando nessas relações, para que uma ferramenta computacional possa gerar listas com a frequência das palavras e, a partir disso, possa executar os cálculos estatísticos que determinam a relevância dos elementos linguísticos, é necessário, em um primeiro momento, que a ferramenta identifique cada um dos elementos linguísticos presentes num texto. Essa identificação é realizada através do processo computacional conhecido na área de PLN como tokenização.

A tokenização pode ser definida como a segmentação das sentenças de um texto em elementos significativos, chamados *tokens*[57], por meio de tokenizadores

[57] Segundo Baker, Hardie e Mcenery (2006), um *token* é uma unidade linguística mínima que, geralmente, corresponde a uma palavra do texto. O tamanho de um *corpus* é medido pela quantidade de *tokens* (palavras) que ele possui.

(*tokenizers*). Consoante Schmid (2008), o isolamento dos *tokens*, geralmente, é feito com base nos espaços e na pontuação existentes entre os caracteres do texto:

> Nas línguas alfabéticas, as palavras são cercadas por espaços em branco e, opcionalmente, precedidas e seguidas por sinais de pontuação, parênteses ou aspas. As sentenças, geralmente, terminam com um ponto final (.), um ponto de interrogação (?) ou um ponto de exclamação (!). A regra básica da tokenização estabelece que a obtenção de um *token* se dá por meio da divisão de uma sequência de caracteres, utilizando-se seus espaços em branco como referência e desconsiderando-se os sinais de pontuação, os parênteses e as aspas que possam estar presentes nas suas extremidades. A regra estabelece, ainda, que as ocorrências de ".", "?" ou "!" delimitam as sentenças. Esta regra básica é bastante precisa porque o espaço em branco e a pontuação são indicadores bastante confiáveis de limites de palavras e sentenças (SCHMID, 2008. p. 529)[58].

Em face dessa citação, poderíamos concluir que a tokenização é um procedimento simples. Todavia, Grefenstette e Tapainainen (1994) afirmam que ela está longe de ser trivial em virtude de envolver escolhas difíceis que podem refletir nos resultados finais de uma pesquisa. Gries (2009) corrobora o ponto de vista de Grefenstette e Tapainainen (1994), no trecho a seguir, ao descrever exemplos de questionamentos e escolhas que podem surgir durante a tokenização de um texto.

> As listas (de frequência) dependem bastante do que você considera como uma palavra e lembre-se de que agora estamos falando sobre o que um computador considera uma palavra. Como o computador somente é capaz de processar cadeias de caracteres, precisamos estabelecer critérios para que ele possa identificá-las em um processo conhecido como tokenização. Nesse sentido, você pode querer definir para o computador que uma palavra é 'uma sequência de letras cercada por espaços'. Bem, tal definição resolve o problema a maior parte do tempo. Porém, às vezes, uma palavra não é seguida por um espaço, mas por um sinal de pontuação. Certo! Então, você poderia dizer que uma palavra é 'uma sequência de letras cercada por espaços ou sinais de pontuação'. E quanto aos casos como Ph.D. ou Dr.? Um colchete é um sinal de pontuação? E quanto aos hifens? Podemos dizer que '*ill-defined*' são duas palavras? Como lidamos com a variação ortográfica? Podemos dizer que '*armchair-linguist*' e '*armchair linguist*' são as mesmas palavras? E '*favour*' e '*favor*'? A sequência '*John's book*' corresponde a duas ou três palavras? Se

[58] Original: "*In alphabetic languages, words are surrounded by whitespace and optionally preceded and followed by punctuation marks, parentheses, or quotes. Sentences usually end with a period (.), a question mark (?) or an exclamation mark (!). A simple tokenization rule can be stated as follows: split the character sequence at whitespace positions and cut off punctuation marks, parentheses, and quotes at both ends of the fragments to obtain the sequence of tokens. Insert a sentence boundary after any occurrence of ".", "?", or "!". This simple rule is quite accurate because whitespace and punctuation are fairly reliable indicators of word and sentence boundaries*".

> considerar que '*John's book*' tem duas palavras, você poderá ser tentado a dizer que uma palavra é 'uma sequência de letras e/ou hifens e/ou apóstrofos cercados por espaços ou sinais de pontuação'. Porém, o que dizer sobre '*he's going'* ou '*isn't it*'? Correspondem a três palavras cada? Caso responda que sim, perceba que o apóstrofo de '*isn't*' nem mesmo corresponde ao lugar em que devemos dividir as palavras. Para complicar ainda mais: '1960' seria uma palavra? Quantas palavras há em '*25-year-old man*'? E quanto a *links* como '*http://www.linguistics.ucsb.edu*'? Imagino que já seja óbvio: está longe de ser claro como definir uma palavra para um computador. E nem cheguei a mencionar os problemas que surgem quando olhamos para outros idiomas além do inglês – como você trataria uma expressão russa escrita em fontes cirílicas em meio a um texto de um jornal escrito em língua inglesa? Os programadores definem as palavras de modos distintos e, como resultado, as diferentes ferramentas de *corpus* geram listas de frequência dessemelhantes para um mesmo *corpus*. As listas de frequência geradas para um mesmo *corpus* podem, ainda, ser díspares quando geradas por versões diferentes de um mesmo programa (GRIES, 2009 p. 1236)[59].

Levando em consideração as perspectivas de Schmid (2008) e Grefenstette e Tapainainen (1994), acreditamos que a pontuação e o espaçamento dos caracteres de um texto exercem um papel fundamental para a delimitação correta dos *tokens* existentes nele. Ao processar um texto, o computador não consegue distinguir o ruído linguístico dos dados linguísticos reais no fluxo de caracteres. Por isso, a tokenização de um texto que apresenta ruído linguístico pode acarretar a presença de *tokens* sem nenhuma relação com qualquer elemento significativo da língua (por exemplo: *tokens* formados por partes de palavras que foram separadas incorretamente) e, consequentemente,

---

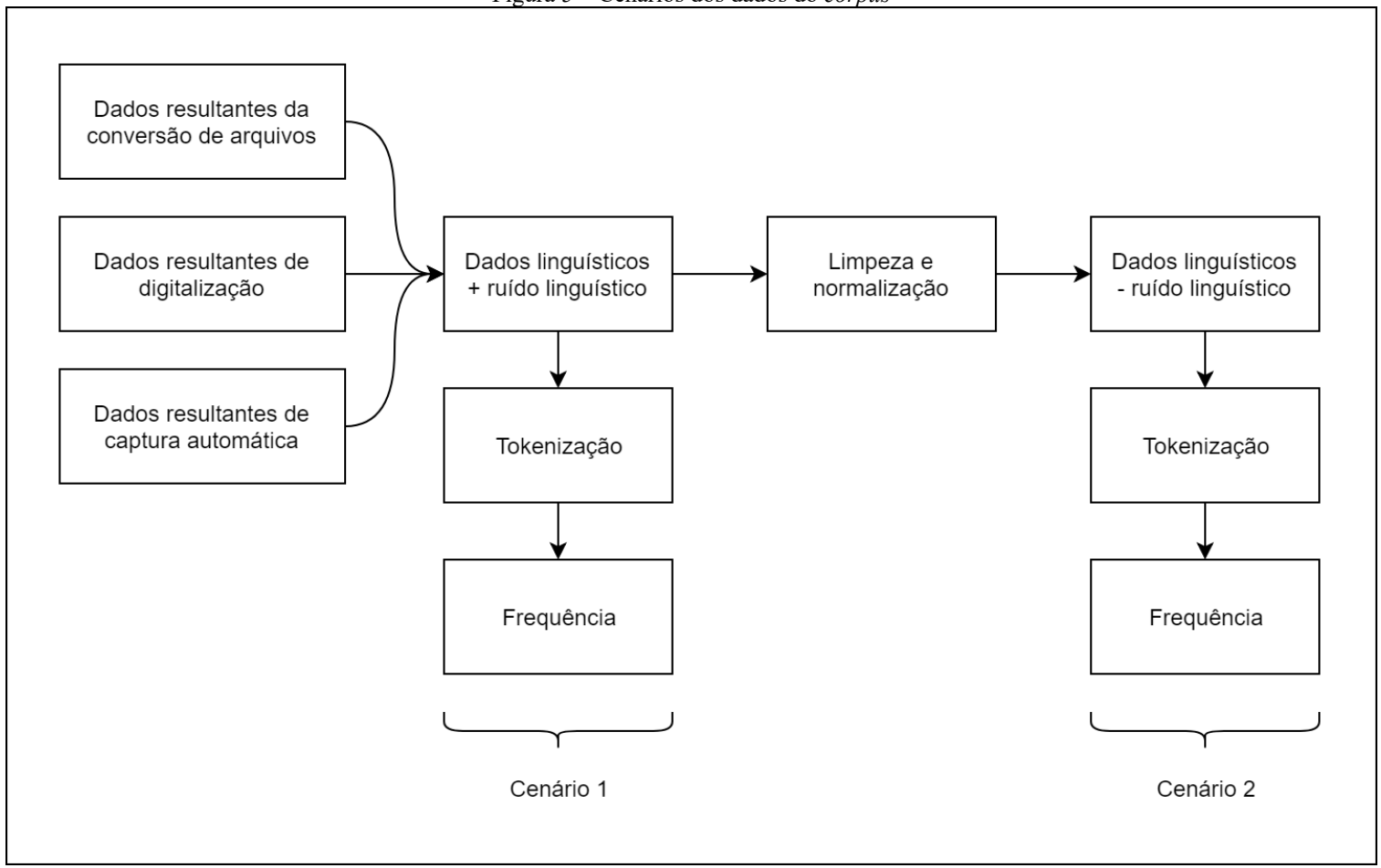
[59] Original: "*Now such lists depend very much on what you think a word is and remember we're now talking about a computer knowing what a word is. The computer only processes strings of characters and must therefore identify all word tokens in the file(s), a process called tokenization. So, you might want to define to a computer what a word as 'a string of letters surrounded by spaces'. Well, that does the trick most of the time, but sometimes a word is not followed by a space, but by a punctuation mark. Ok, so you might say a word is 'a string of letters surrounded by spaces or punctuation marks'. Then what about Ph.D. or just Dr.? Is a bracket a punctuation mark? What about hyphens? Do we want to say ill-defined is two words? How do we handle spelling variation? Do we want to be able to say that armchair-linguist and armchair linguist are the same words? And favour and favor? Is John's book two or three words? If you think John's book is two words, you might then be tempted to say a word is 'a string of letters and / or hyphens and / or apostrophes surrounded by spaces or punctuation marks.' But then what about he's going or isn't it? Aren't these three words each? And if you say 'yes', then note that the apostrophe in isn't it is not even where we would split isn't it up into words: we would say the 'words' are 'is', 'n't', and 'it', not 'isn', 't', and 'it'. Let me make it worse for you: isn't 1960 a word, and how many words does a 25-year-old man contain, or a link such as http://www. linguistics.ucsb.edu? I guess it's obvious by now: it's far from clear how to define to a computer what a word is. And I haven't even mentioned the issues that arise when you look at languages other than English – how do you treat a Russian expression in Cyrillic fonts such as 'Чайковский' in an American newspaper text? Programmers make different distinctions and as a result different corpus software will output different frequency lists for the same corpus, and a corpus program whose definition was changed will output a frequency list for a corpus that is different from a frequency list for the same corpus made with an earlier version of the same program*"

cálculos imprecisos sobre a frequência de um elemento, o que pode comprometer a qualidade das listas provenientes da análise realizada por ferramentas computacionais.

A limpeza e a normalização dos textos de um *corpus* são maneiras de reduzir ou de eliminar erros de tokenização. A primeira, de acordo com Aluísio e Almeida (2006), consiste na remoção dos ruídos linguísticos e a segunda remete-se à correção de erros e padronização de dados linguísticos. A retificação de erros, geralmente, está ligada às palavras que passaram a apresentar a grafia incorreta após a conversão do texto para o formato TXT, e a normalização de dados linguísticos abrange a uniformização de palavras (em termos ortográficos), de siglas e de abreviaturas que possuem variações de escrita, a remoção de espaçamentos e de quebras de linhas desnecessários e a homogeneização de caracteres de pontuação do texto, como hifens, traços, aspas e apóstrofos.

No processo de preparação dos dados de um *corpus*, conforme a necessidade da pesquisa (que pode conter critérios rígidos ou não em relação à qualidade dos dados do *corpus*), dos recursos e da disponibilidade de tempo das pessoas envolvidas na investigação científica, o pesquisador pode optar por realizar ou não a limpeza e a normalização dos textos. De acordo com a escolha realizada, pode surgir um cenário em que o *corpus* é formado tanto pelos dados linguísticos desejáveis quanto pelos indesejáveis (ruído linguístico) ou um quadro em que o *corpus* é composto somente pelos dados linguísticos desejáveis – as duas situações são esquematizadas na Figura 5.

Figura 5 – Cenários dos dados do *corpus*



Fonte: o autor.

Além de ser importante para a criação das listas de palavras, a tokenização é um passo preparatório para a inserção automática de anotações no *corpus* (HANSEN-SCHIRRA, 2003, p. 290), que consiste no acréscimo de marcas com informações adicionais sobre os textos. A anotação é realizada durante a atividade de enriquecimento dos dados, que pode fazer parte do processo de preparação do *corpus*.

De acordo com Almeida (2002), o uso de marcas em textos eletrônicos teve origem na forma como os textos impressos eram codificados. Para o autor, nos textos impressos, o uso de recursos, como sinais de pontuação, letras maiúsculas e minúsculas, espaço entre as palavras e regras para a disposição do texto na página, são tipos de codificação ou de "marcação" que servem para ajudar o leitor a identificar os elementos de conteúdo e estrutura do texto. Com o surgimento dos computadores, o termo "marcação" foi transposto para o contexto de produção de textos eletrônicos com o propósito de referenciar os conjuntos de códigos, convencionados nas "linguagens de marcação", que definem o modo como os conteúdos de um texto devem ser interpretados ou apresentados pelos computadores.

A LC beneficia-se da possibilidade de acrescentar tais "marcações" no momento em que o pesquisador pretende enriquecer os textos puros de um *corpus* por meio da inserção de etiquetas (*tags*) com informações linguísticas (LEECH, 2005, p. 25) ou descritivas sobre o texto – atividade conhecida, na literatura, como etiquetagem do *corpus*. Cabe destacar que a inclusão de etiquetas pode assumir diferentes formas que enfatizam os aspectos linguísticos que se deseja estudar (MARTÍNEZ, 2017, p. 92) tendo em vista os objetivos da pesquisa.

O Quadro 3 apresenta breves descrições acerca dos principais tipos de etiquetagem presentes na literatura da LC.

Quadro 3 – Tipos de etiquetagem da LC

Tipo	**Função / Características**
Parte do discurso (*Part-of-speech* – POS)	É o tipo mais comum (EDWARD, 2015, p. 39) de etiquetagem e consiste na inserção de uma etiqueta para cada palavra do texto com a designação de sua classe gramatical ou classe de palavras (por exemplo, na língua portuguesa: substantivo, verbo, artigo, adjetivo, preposição, numeral, pronome, advérbio, conjunção e interjeição) com base no seu comportamento sintático ou morfológico.

Etiquetagem semântica (*Semantic annotation*)	Alude à inclusão de etiquetas que denotam as características semânticas de uma palavra, servem para sua desambiguação (*word sense disambiguation*) e para a detecção de relações semânticas (por exemplo: sinonímia, hiperonímia e antonímia) com outras palavras do *corpus* (MARTÍNEZ, 2017, p. 127). Para ilustrar: a inserção de uma etiqueta de contexto social para a palavra "dama" distingue seu sentido em relação ao uso dela no contexto de jogos.
Etiquetagem prosódica (*Prosodic annotation*)	Diz respeito ao acréscimo de etiquetas que marquem fenômenos prosódicos (tais como: entonação, pausa, organização temporal) que podem trazer mudanças de sentido para um segmento linguístico (MARTÍNEZ, 2017, p. 117).
Etiquetagem fonética *(Phonetic annotation)*	Consiste na inserção de etiquetas para a descrição da percepção dos sons emitidos no pronunciamento de palavras (MARTÍNEZ, 2017, p. 117).
Etiquetagem fonológica (*Phonologic annotation*)	Remete-se à inclusão de etiquetas para a descrição das funções dos sons e dos seus valores distintivos em relação a outros sons da língua (MARTÍNEZ, 2017, p. 117).
Etiquetagem pragmática (*Pragmatic annotation*)	Reporta-se ao acréscimo de etiquetas para a identificação das intenções de atos de fala ou polaridade (MARTÍNEZ, 2017, p. 131). Exemplos: uma etiqueta pode marcar se um seguimento trata da realização de uma pergunta ou de uma afirmação ou, ainda, se denota uma opinião positiva ou negativa.
Etiquetagem estilística (*Stylistic annotation*)	Alude à inclusão de etiquetas para a identificação de marcas de estilo no texto (LEECH, 2005, p. 26) e, na maioria das vezes, associa-se aos textos literários. Exemplo: etiquetas distintas podem evidenciar pontos do texto em que o discurso direto e o indireto são usados.
Etiquetagem sintática (*Syntactic annotation ou Parsing*)	Consiste na inserção de etiquetas com informações sobre as relações sintáticas, de acordo com uma gramática formal, entre os elementos que constituem uma sentença.
Etiquetagem de erros (*Error-tagging*)	Diz respeito à colocação de etiquetas com a indicação de erros cometidos por aprendizes de segunda língua (LEECH, 2005, p. 26).

Fonte: o autor.

A introdução de etiquetas em um texto pode ser realizada de forma manual, automática ou semiautomática. Nos dois últimos casos, o pesquisador faz o uso dos *softwares* conhecidos como etiquetadores (*taggers*)[60] (MYER, 2004, p. 86), que analisam os textos de um *corpus* e adicionam etiquetas nele em consonância com um conjunto de etiquetas (*tagset*)[61] de que dispõem. Myer (2004) menciona que os etiquetadores podem ser baseados em regras (*rule-based taggers*) ou em probabilidades (*probabilistic taggers*). Consoante o autor, os primeiros fundamentam-se em normas gramaticais e em bases de palavras[62] da língua incorporada à sua programação, e os últimos, em probabilidades estatísticas de ocorrência de uma palavra em determinado contexto.

As etiquetas de cunho descritivo, geralmente, são utilizadas na criação dos cabeçalhos dos arquivos de texto de um *corpus*. Os cabeçalhos são estruturas informacionais extralinguísticas (por exemplo, dados bibliográficos) incluídas no início de um texto. O *tagset* do cabeçalho pode ser construído conforme padrões preestabelecidos, como o *Text Encoding Initiative* (TEI)[63], ou de forma personalizada pelo próprio pesquisador. Em qualquer situação, Berber Sardinha (2004, p. 94) sugere que a definição das etiquetas do cabeçalho deve ser feita levando em consideração as necessidades da pesquisa e a restrição de uso do *corpus* (se existe a pretensão de compartilhamento do *corpus* ou não).

A estrutura dos cabeçalhos é habitualmente implementada com o uso da linguagem de programação XML. Nos padrões da XML, os elementos informacionais são organizados por meio de etiquetas hierárquicas, delimitadas por parênteses angulares, e podem estabelecer uma relação de parentesco entre si, conforme a estrutura apresentada na Figura 6:

Figura 6 – XML *Tree Structure*

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Fonte: *w3schools*

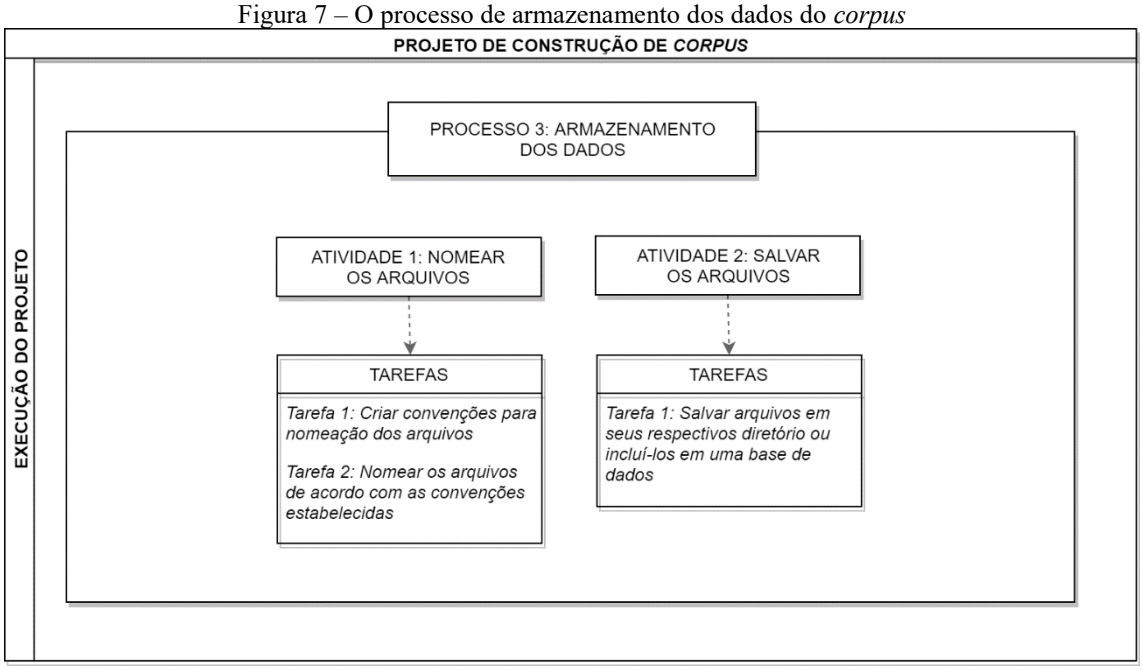
[60] No caso da etiquetagem sintática, os *softwares* utilizados são conhecidos como *parsers*.
[61] De acordo com Myer (2004), existem vários tipos de *tagsets* que se diferenciam pelo número e pelo tipo de suas etiquetas.
[62] Myer (2004) refere-se às bases de palavras como "*program's lexicon*" (MYER, 2004, p. 88).
[63] Disponível em: http://www.tei-c.org. Acesso em: 21 jun. 2018.

O enriquecimento de *corpus* com dados linguísticos e extralinguísticos expande as possibilidades de pesquisa que podem ser realizadas em seu conteúdo ou, nas palavras de Hardie (2012), aperfeiçoa a sua "pesquisabilidade" (*searchability*) (HARDIE, 2012, p. 268). Isso ocorre porque os modos de realizar pesquisas e análises no *corpus* ampliam-se, sendo permitido efetuar buscas tanto a partir de palavras quanto de etiquetas. Nessa perspectiva, o pesquisador poderia, por exemplo, lançar mão do uso de etiquetas como parâmetro de busca para a criação de linhas de concordância nos programas concordanciadores. Além da melhoria na pesquisabilidade do *corpus*, a etiquetagem "auxilia na desambiguação lexical e permite a descrição de padrões léxico-gramaticais" (SARDINHA, 2004, p. 145) e, de acordo com Edward (2015), serve como base para o treinamento de programas de etiquetagem de *corpus* e de *parsing*.

2.4.2.3 O processo de armazenamento dos dados do corpus

Após a obtenção e preparação do *corpus*, os arquivos de texto que o constituem precisam ser nomeados e armazenados de modo que possam ser recuperados facilmente (NELSON, 2010, p. 63). As atividades relacionadas à nomeação, ao armazenamento e à disponibilização dos dados de um *corpus* são realizadas no processo de armazenamento de dados, ilustrado na Figura 7.

Figura 7 – O processo de armazenamento dos dados do *corpus*



Fonte: o autor.

Antes de aprofundarmos nos aspectos desse processo, é interessante apresentarmos alguns conceitos sobre organização de informações pelo homem. Segundo Sedlar (2005), os seres humanos tendem a organizar as informações em categorias que se relacionam entre si de forma hierárquica. O autor toma como exemplo a classificação básica dos seres vivos, na qual os indivíduos são diferenciados por categorias (espécie, gênero, família, ordem e classe).

Para Sedlar (2005), com o advento dos computadores, as técnicas relativas ao gerenciamento das informações eletrônicas passaram a refletir a tendência humana pelas formas de organização hierárquica. Em virtude disso, ele considera que os sistemas operacionais dos computadores foram desenhados com base em princípios conhecidos como "sistemas de arquivos", em que documentos são armazenados em pastas (diretórios) ligadas entre si, de maneira hierárquica, a partir de uma relação intuitiva entre os conteúdos existentes nelas e as categorias que lhes foram associadas. Essa forma de dispor arquivos constitui o que chamamos de "paradigma tradicional de armazenamento de dados" e contrasta com o modelo dos "bancos de dados relacionais", proposto por Edgar Frank Codd, em 1970, no artigo "*A Relational Model of Data for Large Shared Data Banks*".

No modelo de Codd (1970), as informações são armazenadas em múltiplas tabelas (estruturas de linhas e colunas) de um banco de dados conectadas entre si por referências em comum e, segundo O'Regan (2008), podem ser recuperadas com um Sistema Gerenciador de Banco de Dados Relacional (*Relational Database Management System* – RDBMS), que envia consultas construídas na linguagem *Structured Query Language* (SQL) para o banco de dados, recebe o resultado da consulta e o exibe para o usuário. Para Sedlar (2005), o paradigma tradicional de armazenamento de dados tem a vantagem de ser simples, intuitivo e fácil de ser implementado, mas não possui a flexibilidade dos bancos de dados relacionais para a recuperação de dados, conforme o autor destaca no trecho, a seguir:

> Infelizmente, a simplicidade da organização hierárquica não oferece o suporte necessário para operações complexas de recuperação de dados. Por exemplo, o conteúdo de cada diretório pode ter que ser inspecionado para recuperar todos os documentos criados em um determinado dia que tenham um nome de arquivo específico. Como todos os diretórios precisam ser pesquisados, a organização hierárquica não contribui em nada para a facilitação do processo de

> recuperação. Por outro lado, um sistema de banco de dados relacional é adequado para o armazenamento de grandes quantidades de informações e para acessar dados de maneira muito flexível. Ao contrário do que ocorre nos sistemas organizados hierarquicamente, dados que correspondem a critérios de pesquisa complexos podem ser recuperados com facilidade e eficiência nos sistemas de banco de dados relacionais. No entanto, a formulação e o envio de consultas para um servidor de banco de dados são menos intuitivos do que, simplesmente, percorrer uma hierarquia de diretórios e está além do nível de conforto técnico de muitos usuários de computador (SEDLAR, 2005, p. 2)[64].

Apesar da flexibilidade na recuperação de dados oferecida pelos bancos de dados relacionais, podemos afirmar que a barreira técnica para a sua apropriação, o desconhecimento da sua existência como forma de armazenamento de dados, a falta de necessidade por consultas sofisticadas na construção de um *corpus* e a conveniência das práticas tradicionais de armazenamento de dados afastam a maior parte dos linguistas da possibilidade de adotar os bancos de dados relacionais em seus projetos de construção de *corpus*. Acreditamos, ainda, que o fato de as ferramentas computacionais da LC requererem que os dados linguísticos de um *corpus* estejam em um formato de arquivo eletrônico (TXT) direciona os linguistas à adoção do método convencional de armazenamento de arquivos. Portanto, o armazenamento de um *corpus*, comumente, dá-se por meio do salvamento dos arquivos de texto em diretórios organizados de acordo com a estrutura hierárquica adotada no projeto de construção do *corpus*, de modo que os conteúdos dos textos possuam uma associação significativa com a categoria de seu respectivo diretório.

Outra atividade importante é a nomeação dos arquivos de texto para o armazenamento, a manipulação e a recuperação das informações de um *corpus*. Ao criar e salvar um documento eletrônico, o pesquisador precisa atribuir um nome ao arquivo (*filename*) a fim de que ele possa ser identificado no sistema de arquivos do sistema operacional. Em termos computacionais, o nome de um arquivo é uma sequência (*string*) de caracteres atribuída a ele, com a função de permitir que o computador

---

[64] Original: "*Unfortunately, the simplicity of the hierarchical organization does not provide the support required for complex data retrieval operations. For example, the contents of every directory may have to be inspected to retrieve all documents created on a particular day that have a particular filename. Since all directories must be searched, the hierarchical organization does nothing to facilitate the retrieval process. A relational database system is well suited for storing large amounts of information and for accessing data in a very flexible manner. Relative to hierarchically organized systems, data that matches even complex search criteria may be easily and efficiently retrieved from a relational database system. However, the process of formulating and submitting queries to a database server is less intuitive than merely traversing a hierarchy of directories, and is beyond the technical comfort level of many computer users*".

distinga um arquivo de outros e de possibilitar aos usuários do sistema ou ao próprio sistema a associação de um nome descritivo ao documento eletrônico. Para Cooper *et al*. (1995), a atribuição de nomes de arquivos feita pelos seres humanos ocorre, com frequência, por associações mnemônicas que estejam ligadas ao conteúdo deles.

Os nomes de arquivos, em todos os sistemas computacionais, por convenção, são compostos por duas partes separadas por um ponto: a primeira parte (*base file name*) apresenta a nomeação, propriamente dita, conferida a eles, e a segunda remete-se à extensão do arquivo, responsável por identificar o seu formato. Por mais que estabeleça um padrão mínimo, essa convenção, por si só, não oferece critérios suficientes para garantir a relação do nome do arquivo ao seu conteúdo. Diante disso, os seres humanos, nos diversos campos de atuação em que há o uso de documentos eletrônicos, criam convenções mais específicas para a nomeação de arquivos, conhecidas como *File Naming Conventions* (FNC)[65] na área de Gerenciamento de Dados, especialmente voltadas para a primeira parte do nome de um arquivo.

A convenção de nomeação de arquivos pode ser definida como um conjunto de regras que determina a estrutura da nomeação – constituída por diferentes segmentos que abrigam elementos informativos, ou seja, aqueles que fazem referência ao conteúdo, à descrição, ao contexto ou ao propósito dos arquivos. Tais partes da nomeação podem incluir sobre o arquivo: a) data de criação; b) código de identificação (ID); c) nome do autor; d) área do conteúdo; e) versão, entre outros.

A ideia central de convencionar a nomeação é combinar informações suficientes para que a identificação do conteúdo do arquivo seja feita a partir de seu nome. Para exemplificarmos, se adotamos a presença dos seguintes elementos informativos: [Tipo textual]-[Domínio]-[Língua]-[Autor]-[Identificador], um arquivo de um *corpus* poderia apresentar o nome "artigo-linguistica-linguainglesa-oliveira-001.txt". Isso poderia levar o pesquisador a fazer a seguinte inferência (associação): o arquivo trata de um texto em forma de artigo, da área da Linguística, escrito em língua inglesa por um autor chamado Oliveira e é o primeiro de uma sequência de textos coletados.

A relevância das convenções de nomeação de arquivos em projetos de construção de *corpus* é maximizada quando elas são elaboradas a partir de metadados dos textos. Isso ocorre porque os metadados presentes nos nomes dos arquivos podem

[65] As convenções de nome de arquivos também podem ser referenciadas como: a) *naming systems*; b) *naming conventions*; c) *naming schemes*; d) *naming models*.

ser utilizados como referência para a criação de subgrupos de textos, ou seja, sub*corpora* de um *corpus* (REPPEN, 2010, p. 33) por meio de pesquisas comuns nos recursos convencionais de gerenciamento de arquivos dos computadores ou no escopo de ferramentas computacionais da LC, como o *Antconc* (ANTHONY, 2019)[66], que não possuem suporte para a realização da filtragem dos textos de um *corpus* através dos metadados dos cabeçalhos existentes neles (BLECHA, 2012, p. 84). Outrossim, as convenções de nomeação de arquivos reduzem o esforço para a compilação de textos ao oferecerem parâmetros para a nomeação manual e automática dos arquivos.

2.4.3 A fase de encerramento do projeto

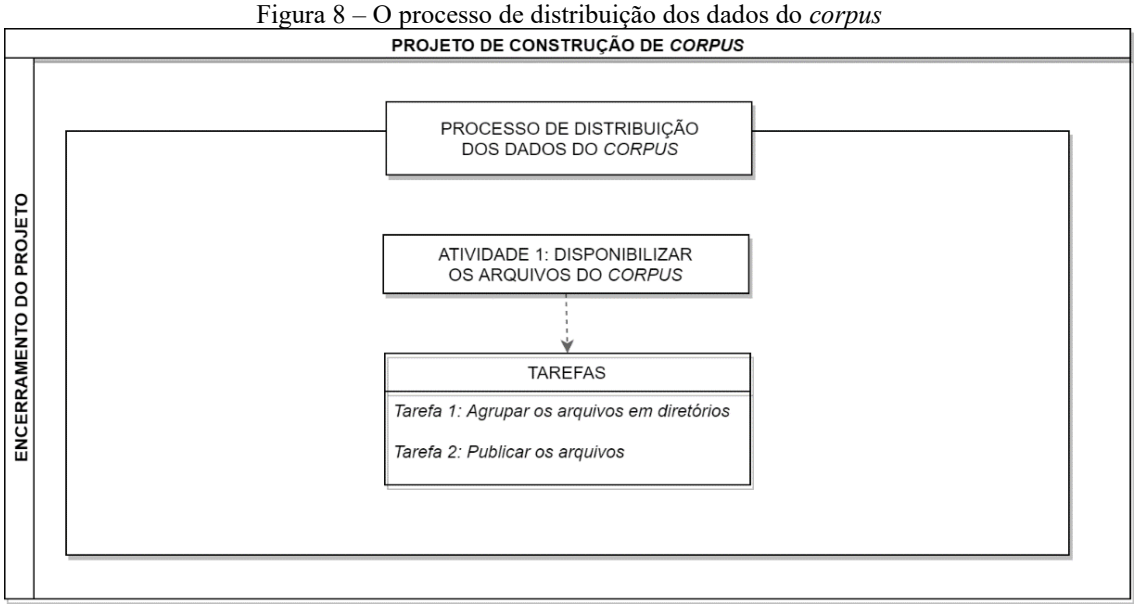
A fase de encerramento é a última do projeto de construção manual de um *corpus* e caracteriza-se pela disponibilização dos seus dados para o processamento por ferramentas computacionais de análise linguística. A disponibilização do *corpus* dá-se pelo processo de distribuição dos seus dados, conforme descrito no próximo tópico.

2.4.3.1 O processo de distribuição dos dados do corpus

Depois das etapas de planejamento e execução, a construção de um *corpus* segue para a fase de encerramento, que compreende o processo de distribuição dos dados do *corpus*, ilustrado na Figura 8.

[66] Disponível em: http://www.laurenceanthony.net/software/antconc/. Acesso em: 6 mar. 2019.

Figura 8 – O processo de distribuição dos dados do *corpus*



Fonte: o autor.

A distribuição de um *corpus* consiste na disponibilização de seus dados com o propósito de serem processados pelas ferramentas computacionais, nas quais os métodos de análises e recuperação de informações são aplicados. Nessa fase, é comum que haja o agrupamento dos arquivos em diretórios que refletem a estrutura hierárquica do projeto do *corpus* e a posterior publicação dele em um meio acessível (um dispositivo de armazenamento de dados ou um servidor de Internet) para os seus usuários finais.

2.5 O tempo e o esforço na construção de um *corpus*

Antes do surgimento da Internet, a realização de pesquisas com *corpora* eletrônicos demandava muito tempo e esforço por parte do pesquisador, pois a construção deles dependia da obtenção dos dados linguísticos em documentos impressos e da conversão destes para o formato eletrônico por meio da digitação[67] ou da digitalização (MCENERY; HARDIE, 2011, p. 57). Ademais, os métodos simples de análise de texto, como a extração de listas de palavras e de seus contextos imediatos, estavam sujeitos a um trabalho realizado manualmente (EDWARD, 2015, p. 32).

A introdução dos computadores no fazer linguístico reduziu a duração das atividades relacionadas à análise de texto e o esforço humano investido nelas, e a

[67] Para Myer (2004), a dificuldade em tornar os textos eletrônicos por meio de procedimentos como a digitação é uma das explicações para o tamanho pequeno dos primeiros *corpora*. O autor cita o exemplo da construção do *Brown Corpus*, em que os textos tiveram de ser digitados à mão.

disponibilização crescente de informações em formato eletrônico na Internet facilitou a obtenção de dados linguísticos para a construção de *corpora*. Porém, na literatura da LC, não é raro encontrarmos autores (cf. ATKINS; CLEAR; OSTLER, 1992, p. 4; BAKER, 2010, p. 109; BIANCHI, 2012, p. 36; EDWARD, 2015, p. 36; EVANS, 2007, n. p; KÜBLER; ASTON, 2010, p. 512; MACMULLEN, 2003, p. 15; MCENERY; HARDIE, 2011, p. 4; MCENERY; XIAO; TONO, 2006, n. p; MINSHALL, 2013, p. 20; RENOUF, 2007, p. 42; SEMINO; SHORT, 2004, p. 226; VOORMANN; GUT, 2008, p. 237; ZANETTIN, 2014, p. 32) que afirmem que as atividades relacionadas à análise e à construção de *corpora* eletrônicos ainda podem requerer quantidades enormes de tempo e esforço.

De fato, a construção manual de *corpora*, em especial, ainda pode exigir bastante esforço e tempo do pesquisador por: a) envolver um conjunto complexo de atividades dos processos de planejamento (cf. Figura 2) e obtenção do *corpus* (cf. Figura 3); preparação (cf. Figura 4), armazenamento (cf. Figura 7) e distribuição de dados do *corpus* (cf. Figura 8); b) incluir dados que podem variar em número e qualidade (ANTHONY, 2013, p. 144); c) necessitar de constante intervenção manual por parte do pesquisador; d) não contar com recursos computacionais para a automatização de algumas atividades.

Dentre as referidas questões que podem influenciar o tempo e o esforço para a elaboração manual do *corpus*, a intervenção do pesquisador ganha destaque ao estar presente em praticamente todas as fases do projeto (BAKER, 2010, p. 109), conforme descrevemos em cada uma das atividades enumeradas a seguir:

1) **Definir o desenho do *corpus*, os recursos utilizados para lidar com ele e o cronograma referente ao projeto de construção do *corpus*** (Fase inicial/Processo de planejamento do *corpus*)**:** faz com que o pesquisador, basicamente, tome decisões teóricas e gerenciais.

2) **Localizar dados linguísticos** (Fase de execução/Processo de obtenção dos dados do *corpus)***:** a coleta manual de textos, em oposição à automática feita pelos *web crawlers*[68], exige que o pesquisador busque

[68] Segundo Najork e Heydon (2002), os *web crawlers* são programas que encontram e fazem o *download* automático de documentos da Internet. Os autores explicam que, ao localizarem um documento, os *web*

fontes e verifique o acesso ao material (eletrônico, impresso ou em outro estado) que será utilizado em sua seleção. Os dados em formato eletrônico, por exemplo, são frequentemente localizados por meio de pesquisas feitas em sistemas de busca como o *Google*. Mesmo com a facilidade oferecida por esse tipo de sistema, a filtragem dos materiais encontrados (feita com base nos critérios do desenho do *corpus*) pode ser árdua devido ao grande volume e à qualidade das informações retornadas por *sites* como *Google*. A título de exemplificação, pesquisamos a expressão "inteligência artificial" (entre aspas) no *Google* e obtivemos, aproximadamente, 4.700.000 resultados. Após o refinamento da busca com a inclusão do operador de pesquisa[69] *filetype:pdf*, o sistema retornou em torno de 539.000 resultados. Podemos dizer que, por mais que o refinamento da busca tenha reduzido a quantidade de resultados, a seleção dos possíveis textos que poderiam ser acrescentados a um *corpus* seria difícil.

3) **Obter permissão de uso dos dados linguísticos** (Fase de execução/Processo de obtenção dos dados do *corpus*)**:** com base em Santos (2011), essa atividade implica que o pesquisador precisará identificar a pessoa ou a entidade detentora dos direitos autorais de um texto, solicitar-lhe consentimento para usá-lo e, em seguida, aguardar retorno. Conforme Santos (2011), o consumo de tempo é ampliado nos casos em que o pesquisador precisa realizar várias tentativas de contato com o detentor para ter sucesso ou nas situações em que a solicitação de autorização tenha de partir de níveis hierárquicos superiores de uma instituição para que se obtenha uma resposta.

4) **Capturar os dados** (Fase de execução/Processo de obtenção dos dados do *corpus*)**:** demanda a intervenção humana em uma escala que varia de acordo com o formato em que os dados estão quando são encontrados. Se estiverem no eletrônico, o pesquisador necessitará de intervir menos, já

---

crawlers extraem os endereços (URLs) contidos nele e os utilizam para continuar a pesquisa por outros documentos de forma indefinida ou até que uma condição preestabelecida seja atingida.

[69] Os operadores de pesquisa, de acordo com o suporte da *Google* (2019), são símbolos ou palavras adicionadas nas expressões de pesquisa para tornar os resultados mais precisos.

que sua tarefa, normalmente, será a de realizar o *download* de arquivos. Se os dados estiverem em materiais impressos ou escritos à mão, o pesquisador terá de convertê-los para o formato eletrônico, de preferência, por meio da digitalização com o auxílio de *scanners* e *softwares* de OCR – uma das atividades que mais demandam esforço e tempo. Para Simske (2006), a precisão oferecida atualmente pelos OCRs na conversão de textos ainda é limitada e pode gerar erros referentes à troca, à inserção e à exclusão de caracteres, principalmente, quando a qualidade dos documentos originais é ruim. Isso faz com que autores como Nelson (2010), Kübler e Aston (2010), Santos (2011) e Bianchi (2012) preconizem a revisão manual cuidadosa dos dados resultantes da digitalização de textos por intermédio de *scanners* e OCRs para que se tenha certeza de que eles correspondem às suas versões originais. Em um cenário pessimista, segundo Morrison, Popham e Wikander (2019), podemos encontrar textos com a qualidade degradada ao ponto de torná-los inviáveis para o processamento pelos OCRs. Nesses casos, os autores afirmam que o tempo gasto para corrigir os erros de digitalização pode ser maior do que o tempo gasto para a digitação (*keyboarding*) do texto. Assim, a opção mais viável, talvez a única, quando os dados linguísticos estão em textos impressos com má qualidade ou estão escritos à mão é a digitação, que é considerada por Baker, Hardie e Mcenery (2006) e Kennedy (1998) a forma mais demorada de capturar textos.

5) **Converter os dados** (Fase de execução/Processo de preparação dos dados do *corpus*)**:** exige pouco do pesquisador, apenas que ele manipule ferramentas computacionais que convertam os arquivos para o formato TXT e para o padrão *Unicode UFT-8*. Transformar um texto escrito em PDF para TXT, por exemplo, pode ser feito de forma gratuita por meio de serviços *on-line* de conversão, como o *Lightpdf* (lightpdf.com) e o *Pdfcandy* (pdfcandy.com), entre outros. E a mudança para o padrão *Unicode UTF-8* pode ser efetuada por ferramentas como o *EncodeAnt* (ANTHONY, 2016).

6) **Limpar e normalizar os dados** (Fase de execução/Processo de preparação dos dados do *corpus*)**:** requer que o pesquisador proceda como auditor no que diz respeito aos dados do *corpus* para identificação e posterior eliminação ou correção das anomalias (ruído linguístico). A limpeza e a normalização estão diretamente relacionadas a algumas variáveis: volume e qualidade dos dados (resultante dos métodos de captura, conversão e codificação dos textos), finalidade (necessidades) da pesquisa e, por fim, métodos escolhidos para a execução da limpeza e da normalização. Vale lembrar que as tarefas em questão podem ganhar proporções gigantescas e, portanto, serem difíceis no caso de *corpora* compostos por grandes volumes de informação. Segundo Dasu e Johnson (2003), as duas atividades podem ocupar cerca de 80% do tempo compreendido entre a obtenção e a análise de um texto;

7) **Enriquecer os dados** (Fase de execução/Processo de preparação dos dados do *corpus*)**:** pressupõe que o pesquisador realize uma conferência no que alude à etiquetagem automática de *corpora*. Conforme Neumann e Hansen-Schirra (2012), o enriquecimento de *corpora* grandes depende da etiquetagem automática, pois o processamento manual de grandes volumes de dados é praticamente inviável. Semino e Short (2004) reforçam essa ideia ao afirmarem que até mesmo a etiquetagem manual de *corpora* pequenos é extremamente demorada. Contudo, a utilização de ferramentas computacionais para a etiquetagem não dispensa a intervenção manual do pesquisador (MEYER, 2004, p. 140), uma vez que *taggers* e *parsers* não conseguem alcançar uma precisão total no processamento dos dados. Por essa razão, Neumann e Hansen-Schirra (2012) consideram que a confiabilidade[70] de uma etiquetagem está sujeita ao que a ferramenta utilizada pode oferecer. Para Meyer (2004), a precisão dos etiquetadores, geralmente, é comprometida pela inconsistência dos dados (dados não limpos ou normalizados) e pela dificuldade que apresentam para lidar com as características

[70] Segundo Neumann e Hansen-Schirra (2012), a confiabilidade de uma etiquetagem é referente à exatidão (*accuracy*) dos resultados do processamento pela ferramenta computacional. Para Leech (2005), a exatidão de um *tagger* consiste na porcentagem de palavras que ele consegue etiquetar de forma correta.

idiossincráticas (SMITH, 1997, p. 147 *apud* MEYER, 2004, p. 89) da linguagem humana. Em decorrência dos possíveis erros, o resultado da etiquetagem automática precisa ser conferido pelo pesquisador (*post-editing*) com o objetivo de corrigir e resolver possíveis ambiguidades nas etiquetas (LEECH, 2005, p. 38). Segundo Bianchi (2012), essa atividade é desenvolvida manualmente e exige muito tempo e esforço. Por isso, para Nivre (2008, p. 227), durante o desenho de um *corpus*, existe um impasse inevitável entre o volume de dados que irá constituir o *corpus* e a etiquetagem a ser realizada.

8) **Nomear arquivos** (Fase de execução/Processo de armazenamento dos dados do *corpus*)**:** prevê que o pesquisador atribua nomes aos arquivos do *corpus*, de preferência, após estabelecer uma convenção[71]. Nessa atividade, o pesquisador poderá ter de checar como foi definida a estrutura da convenção quando for nomear cada arquivo do *corpus* caso não consiga memorizá-la. Ademais, ele precisará selecionar a informação mais adequada para compor cada segmento da estrutura do nome do arquivo.

9) **Salvar arquivos** (Fase de execução/Processo de armazenamento dos dados do *corpus*)**:** requer pouco do pesquisador quando é feito por meio da alocação dos arquivos em diretórios de um sistema de arquivos, de acordo com a hierarquia estabelecida em um projeto. Entretanto, segundo Sedlar (2005), em situações em que o pesquisador decida salvar os arquivos em uma base de dados, a execução da tarefa dependerá do uso de uma ferramenta computacional que ofereça a interface necessária para a inclusão dos arquivos no banco de dados. O pesquisador poderá optar pelo uso de uma ferramenta já existente ou pela criação de uma ferramenta customizada para o seu projeto. No primeiro caso, ele precisará de um esforço adicional para a escolha de uma ferramenta e para a assimilação do seu uso. No segundo, além do esforço para aprender a usar a ferramenta, ele investirá recursos financeiros, tempo e

[71] A definição de uma convenção, por si só, é uma tarefa que exige esforço e tempo do pesquisador.

esforço por ter de contratar um profissional para desenvolver a aplicação ou por desenvolvê-la por conta própria.

10) **Disponibilizar arquivos** (Fase de encerramento/Processo de distribuição dos dados do *corpus*)**:** demanda pouco esforço e tempo do pesquisador quando ele opta por apenas copiar os arquivos em dispositivos de armazenamento de dados, a saber: *pen drives*, CDs e DVDs. Já a publicação *on-line* do *corpus* (com seus arquivos disponíveis para *download*) pode requerer recursos financeiros (por exemplo, para a contratação de serviços de hospedagem ou de armazenamento de dados na nuvem) e, ainda, mais esforço e tempo do pesquisador, pois ele deverá se preocupar com as seguintes questões: escolha de um local para a publicação, compactação dos arquivos para formatos como o Zip[72], disponibilização de documentação sobre o *corpus* com informações suficientes para que a sua utilização seja feita por outros pesquisadores e explicitação de uma licença de uso dos dados do *corpus*.

O esforço e o tempo necessários para a realização das atividades de construção de *corpora*, somados à disponibilidade de dados na Internet, levaram ao surgimento de *web corpora* ou *corpora ad-hoc*, que são *corpora* compostos por dados coletados da Internet de forma automática. Nesse caso, os linguistas lançam mão de ferramentas computacionais, como o *WebBootCat* (BARONI *et al*., 2006), o *WebCorp Linguist's Search Engine* (KEHOE; GEE, 2007) e o *Bootcat* (BARONI; BERNARDINI, 2004), que, segundo Aluísio e Almeida (2006), utilizam motores de busca (*Google*, por exemplo) e um "pequeno conjunto de itens léxicos, denominados sementes (*seeds*)" (ALUÍSIO; ALMEIDA, 2006, p. 168) para efetuarem a compilação.

Schäfer e Bildhauer (2013) consideram que a realização de inferências estatísticas a partir de *corpora* construídos com base em resultados de pesquisas de motores de busca não é uma boa prática de pesquisa. Eles argumentam que os buscadores privilegiam a precisão (*precision*) em detrimento da revocação (*recall*)[73],

[72] De acordo com *Microsoft* (2016), "os arquivos compactados (zipados) ocupam menos espaço de armazenamento e podem ser transferidos para outros computadores mais rapidamente do que os arquivos descompactados" (*MICROSOFT*, 2016).

[73] Consoante Rubi (2009), a revocação "pode ser mensurada por meio da relação entre o número de documentos relevantes sobre determinado tema, recuperados pelo sistema de busca, e o número total de

podem ser influenciados por fatores econômicos[74], usam variáveis como a língua e a localização de quem fez a pesquisa e realizam alterações automáticas nas expressões fornecidas para a pesquisa (otimizam as expressões por meio de reduções ou expansões). Além disso, as buscas não podem ser reproduzidas devido à constante entrada e saída de conteúdos (indexação) na Internet.

Mais do que as questões relacionadas aos critérios de recuperação de informações dos motores de busca, Schäfer e Bildhauer (2013) acreditam que a opção pelo uso de *corpora* provenientes de métodos automáticos de coleta requer precaução extra do pesquisador no que diz respeito a alguns aspectos, tais como: remoção do *boilerplate*[75] (quais partes do documento foram removidas) e do ruído linguístico dos documentos (quais os tipos de ruídos existentes e qual a precisão da remoção deles); introdução de ruído linguístico (quais ruídos foram introduzidos após o processamento dos documentos); remoção de arquivos duplicados (*deduplication*) (quais documentos foram removidos e quais foram os critérios de remoção) e forma pela qual a amostragem dos dados foi criada.

Em suma, podemos afirmar que, embora exista certa facilidade na compilação automática de *corpora*, essa prática não está isenta de percalços. Com base na literatura da LC, classificamos, a seguir, os problemas que podem surgir numa situação em que se opta por automatizar a coleta de um *corpus*:

1) **Problema da replicabilidade:** está relacionado à mutabilidade dos dados na Internet. Para Mcenery e Hardie (2011), os estudos com *corpora* coletados de forma automática na Internet são difíceis de serem replicados com o passar do tempo em virtude de haver constante mudança de dados na rede;

2) **Problema dos falso-positivos:** os falso-positivos são *tokens* e *types* que não possuem relação com qualquer elemento significativo de uma língua alvo de pesquisa, provenientes de erros de tokenização provocados pelo ruído linguístico de um *corpus*.

---

documentos sobre o tema, existentes nos registros do mesmo sistema" (RUBI, 2009, p. 85). A precisão "pode ser mensurada por meio da relação entre os documentos relevantes recuperados e número total de documentos recuperados" (RUBI, 2009, p. 85-86).

[74] Por exemplo, os conteúdos patrocinados.

[75] Autores como Bianchi (2012) e Bergh e Zanchetta (2008) definem o termo *boilerplate* como partes ou elementos que se repetem entre várias páginas de Internet (BERGH; ZANCHETTA, 2008, p. 320; BIANCHI, 2012, p. 71). Do nosso ponto de vista, a acepção dos autores deveria explicitar que o termo *boilerplate* pode se referir tanto aos dados textuais (por exemplo: as informações de *copyright* e os "rótulos" de elementos de navegação "menus"), quanto aos códigos característicos das linguagens de programação (por exemplo: o HTML5 e o XML) utilizadas na construção das páginas de Internet. No primeiro caso, utiliza-se o termo *boilerplate text* e, no segundo, o termo *boilerplate code*.

Conforme Schäfer e Bildhauer (2013), os *web corpora* tendem a conter um alto nível de ruído mesmo após sua limpeza. Os autores explicam que problemas de pontuação incorreta (omissão de marcas de pontuação e de espaços em branco), limpeza incompleta de marcações estruturais, uso de ortografia não padrão (característica dos gêneros textuais da Internet) e erros de ortografia são comuns nos *web corpora* e podem provocar distorções massivas nas estatísticas de um *corpus*, reduzindo, inclusive, a utilidade dele para pesquisas que dependem de seu tamanho;

3) **Problema da amostragem:** refere-se à incontrolabilidade e arbitrariedade da escolha dos dados dos *web corpora* no universo heterogêneo (RENOUF, 2007, p. 42) dos dados disponíveis na Internet. De acordo com Schäfer e Bildhauer (2013), a coleta automática de documentos, geralmente, não segue um esquema amostral preestabelecido. Em outras palavras, podemos dizer que *web corpora* não são construídos em conformidade com um desenho de *corpus* do método tradicional. Como consequência da ausência do esquema amostral, Schäfer e Bildhauer (2013) afirmam que, na melhor das hipóteses, *web corpora* são uma amostra randômica de dados da Internet, cuja composição exata é desconhecida e precisará ser estabelecida após a sua compilação. Para Mcenery e Hardie (2011), um dos aspectos do desconhecimento do conteúdo de *web corpora* é a dificuldade em determinar o gênero textual dos documentos coletados sem tê-los lido;

4) **Problema legal:** consiste no *download* e uso de textos de *sites* da Internet e na sua distribuição como parte de um *corpus* sem o consentimento dos autores (MCENERY; HARDIE, 2011, p. 58). Segundo Mcenery e Hardie (2011), as leis de direito autoral aplicam-se aos textos coletados automaticamente da Internet do mesmo modo que se aplicam aos materiais impressos e, por isso, podem gerar as mesmas implicações legais que outras formas de construção de *corpora*;

5) **Problema da violação da integridade dos textos:** Schäfer e Bildhauer (2013) argumentam que o processamento automático de coleta dos textos introduz erros nos dados originais deles que podem reduzir a qualidade dos dados. Para exemplificar, os autores mencionam a remoção automática de dados duplicados que, se for feita no interior dos textos, no nível dos parágrafos, pode implicar a inclusão de materiais incompletos em um *corpus*. Consoante Schäfer e Bildhauer (2013), esse tipo de alteração é inaceitável nas pesquisas em que as decisões sobre a seleção dos dados precisam ser feitas de forma consciente pelo pesquisador.

6) **Problema das consultas em massa *(batch or bulk requests*):** a obtenção dos dados dos *web corpora* depende do envio automático, por parte das ferramentas computacionais de coleta automática de dados, das sementes (ALUÍSIO; ALMEIDA, 2006, p. 168) que serão utilizadas como parâmetro de consulta pelos motores de busca. Schäfer e Bildhauer (2013) explicam que, no intuito de evitar abusos, os motores de busca apresentam restrições para o processamento gratuito de grandes volumes de consultas automáticas. Portanto, a construção de *web corpora* de grandes proporções por meio da coleta automática de dados demandará do pesquisador o pagamento pelo serviço de busca que ultrapassa os limites de consultas dos motores até que o volume de dados necessário para os *corpora* seja atingido.

A compilação automática de *corpora* pode ser "adequada para uma grande variedade de propósitos" (MCENERY; HARDIE, 2011, p. 8)[76] e os *web corpora* são extremamente valiosos para as pesquisas que demandam a análise de grandes volumes de informação (BERGH; ZANCHETTA, 2008, p. 320) e em que "o valor do volume dos dados se sobrepõe à qualidade proporcionada pela sua limpeza" (SCHÄFER; BILDHAUER, 2013, p. 126)[77], ainda que apresentem problemas e possam ter sua utilidade considerada limitada por grupos de linguistas (KILGARRIFF; RUNDELL, 2011, p. 262). Para Kilgarriff e Rundell (2011), os *web corpora* adequam-se, por exemplo, às pesquisas lexicográficas para a criação de dicionários gerais em que "os benefícios da abundância de dados superam os problemas dos *web corpora*" (KILGARRIFF; RUNDELL, 2011, p. 262)[78].

Considerados o tempo e o esforço requeridos para a construção manual de *corpora* e os aspectos referentes à compilação automática de *corpora*, Kübler e Aston (2010) sugerem que, antes de se iniciar a empreitada de criação de um *corpus*, seja realizada uma cautelosa investigação sobre a existência de um *corpus* que se adeque às necessidades de uma pesquisa. Por outro lado, Semino e Short (2004) advogam pelo embarque no projeto de elaboração manual de *corpora* ao considerarem que os ganhos da execução deles compensam, amplamente, o tempo e o esforço despendidos pelo pesquisador. Cientes de que os *corpora* são construídos de acordo com o objetivo de cada investigação, sendo, portanto, diferentes uns dos outros, acreditamos que a identificação de um *corpus* já existente que possa ser adotado em uma pesquisa pode

[76] Original: "*suitable for a wide variety of purposes*".
[77] Original: "*values the amount of available data more highly than the cleanliness of a corpus*".
[78] Original: "*the benefits of abundant data outweigh most of the perceived disadvantages of web corpora*".

não ser factível, o que compele o pesquisador a construir, manualmente, seu próprio *corpus* de estudo.

A elaboração manual de *corpora* compostos por volumes pequenos de dados pode demandar pouco tempo e esforço do pesquisador. No entanto, quando o projeto exige a construção manual de *corpora* compostos por grande volume de informação, o tempo e esforço necessários para a sua execução expandem-se na mesma proporção que o tamanho dos *corpora*, tornando-a pouco atrativa.

Além do tempo e do esforço, outro fator que contribui para a falta de atratividade da construção manual de *corpora* é a inexistência de ferramentas computacionais desenhadas especificamente para o suporte às atividades de elaboração de *corpora*. Conforme mencionamos anteriormente, a ausência de suporte para a construção manual de *corpora* leva os pesquisadores a usarem *softwares* tradicionais de gerenciamento e controle de dados, como o utilitário *Windows Explorer*, do sistema operacional *Windows*, e o *Excel*, da *Microsoft*, que consideramos ineficientes, por exigirem grande intervenção manual do pesquisador.

Por essas razões, cabe aos linguistas buscarem formas de acelerar e minimizar o tempo e o esforço necessários para a construção manual de *corpora*. A exemplo do que foi feito em relação à análise de *corpora* (o uso intensivo de computadores para a criação automática de listas de palavras, linhas de concordância – KWIC e evidenciação de padrões linguísticos), à construção dos *web corpora* (a coleta automática de dados) e à etiquetagem de *corpora* (etiquetagem automática por meio de *taggers* e *parsers*), é igualmente importante que as atividades relacionadas à criação manual de *corpora* possam beneficiar-se da automatização proporcionada pelas abordagens computacionais.

3 METODOLOGIA

O objetivo principal desta pesquisa é determinar os efeitos da incorporação do *ToGatherUp* no esforço necessário para a construção manual de *corpora*. A fim de alcançarmos esse propósito, realizamos um experimento de comparação em que observamos os esforços necessários para a construção de duas versões idênticas do CoCLI, sendo que o projeto de elaboração de uma delas contou com a incorporação do *ToGatherUp* e o outro não.

Para que a confrontação fosse possível, em um primeiro momento, estabelecemos um critério objetivo e um método para a medição do esforço das atividades de cada um dos projetos de construção de *corpora*. Na sequência, à medida que executamos a construção dos *corpora*, tabulamos os esforços necessários para a realização de cada uma das atividades dos projetos. Por fim, realizamos um experimento por meio de um teste estatístico para a comparação dos dados tabulados.

Neste capítulo, discorremos sobre os critérios, os métodos e os instrumentos da pesquisa e explicitamos como utilizamos esses recursos para determinar os efeitos da incorporação do *ToGatherUp* nos projetos de construção manual de *corpora*.

Para melhor compreensão do trabalho que efetuamos, em primeiro lugar, explanamos sobre o critério e o método de medição do esforço das atividades de cada um dos projetos de construção de *corpora*; logo após, apresentamos o *ToGatherUp*; e, em seguida, esclarecemos o método usado para construir as duas versões do CoCLI e o instrumento utilizado para a tabulação dos dados referentes ao esforço das atividades dos projetos em questão.

3.1 A medida do esforço

Apesar de o esforço ser um tema recorrente entre os autores da LC (cf. tópico 1.3 Justificativa do primeiro capítulo), na nossa revisão da literatura da área, não identificamos trabalhos que tenham se debruçado sobre a sua investigação. Tendo em vista o resultado dessa averiguação e o objetivo da nossa pesquisa, foi necessário desenvolvermos uma métrica e um método de mensuração do esforço dos projetos de construção de *corpora*.

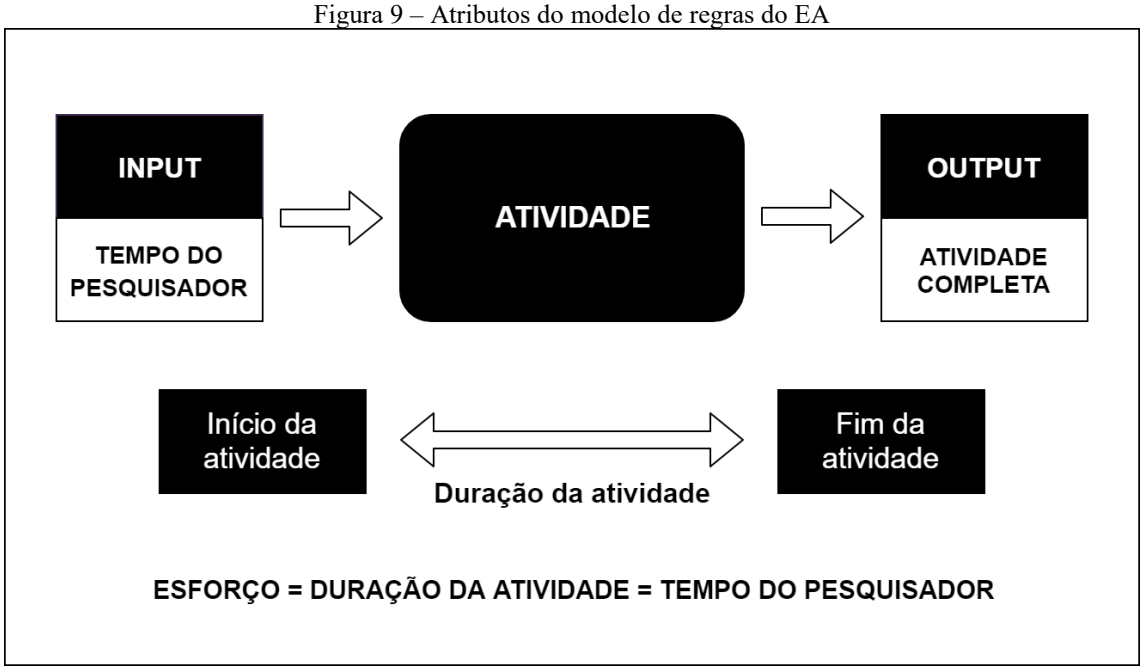
A criação da nossa métrica baseou-se no conceito de medição proposto por Fenton e Bieman (2014), no livro "*Software Metrics: A Rigorous and Practical*

Approach". Conforme os autores, "medição é o processo pelo qual números ou símbolos são associados aos atributos[79] de uma entidade[80] do mundo real, de modo que seja possível descrevê-los de acordo com um conjunto de regras[81] bem definidas[82]" (FENTON; BIEMAN, 2014, p. 5) e compará-los com atributos semelhantes de outras entidades. Em outras palavras, o processo de medição produz medidas que atribuem uma informação, geralmente, em uma escala métrica ou matemática, aos atributos de entidades e permitem que elas sejam descritas e comparadas a partir dessa informação.

Desse modo, assumimos as atividades dos projetos de construção de *corpora* como entidades e consideramos como atributos delas o *input*[83], o *output*[84] e o tempo, realizando a medição deles. Ademais, compusemos um modelo (um conjunto de regras) para a construção do Esforço da Atividade (EA), utilizando a métrica[85] que quantifica, em segundos[86], o esforço despendido para a completude de uma atividade de um projeto de construção de *corpus*. Os atributos que utilizamos e medimos estão ilustrados na Figura 9.

[79] Os atributos são as características ou propriedades das entidades.
[80] As entidades são representações de objetos e eventos do mundo real. Por exemplo: uma pessoa, um lugar, um objeto, uma ideia, um produto, um processo ou uma atividade. Do mesmo modo que uma pessoa (entidade) pode ser descrita a partir de suas características (por exemplo: altura, sexo e idade), as atividades podem ser descritas a partir de seus atributos (por exemplo: duração, *inputs* e *outputs*).
[81] As regras ditam como a medição deve ser realizada.
[82] Original: "*Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way so as to describe them according to clearly defined rules*".
[83] Os *inputs* são as entradas necessárias para a realização de uma atividade. No caso, realizamos um recorte nas entradas que considerou apenas o tempo despendido pelo criador do *corpus* na execução da atividade.
[84] Os *outputs* são os produtos ou entregas (resultados) de uma atividade.
[85] Métricas são unidades de medidas criadas a partir de medições.
[86] Adotamos os segundos para o cálculo das métricas desta pesquisa por serem as unidades básicas de tempo. Além disso, os valores expressos em segundos podem ser convertidos, facilmente, para outras unidades de tempo, como horas e minutos.

Figura 9 – Atributos do modelo de regras do EA



Fonte: o autor.

O primeiro atributo (Atributo 1) de uma atividade é o tempo do pesquisador[87], que corresponde à entrada (*input*), necessário para a realização da atividade. O segundo atributo (Atributo 2) é o resultado da atividade, ou seja, a sua completude[88] e corresponde ao *output*. O Atributo 1 corresponde à quantidade de segundos gastos pelo pesquisador na realização da atividade e o Atributo 2 corresponde ao número 1 (um) – forma que estabelecemos para quantificar e denotar a completude da atividade[89].

O terceiro atributo (Atributo 3) e o quarto atributo (Atributo 4) informam, respectivamente, o início e o fim da atividade e são usados para o cálculo do quinto atributo (Atributo 5), que corresponde à duração da atividade. O Atributo 5 é igual ao intervalo de tempo, em segundos, decorrido entre os atributos 3 e 4. A partir dos atributos mencionados, definimos a regra para o cálculo do EA.

Com base nisso, o EA é igual ao quociente entre o Atributo 5 (a duração da atividade) e o Atributo 2 (a completude da atividade), definido em segundos (unidade de medida), conforme a expressão:

---

[87] Apesar de utilizarmos somente o tempo do pesquisador como *input* da atividade, estamos cientes da existência de outros *inputs* necessários para a realização de uma tarefa, como o conhecimento do pesquisador. A decisão pelo uso do tempo do pesquisador justifica-se pelo fato de o tempo ser, geralmente, reportado como o recurso primário para a execução de uma atividade. Ademais, o tempo do pesquisador apresenta-se como um *input* quantificável e de fácil mensuração em relação aos *inputs* mais abstratos, como o conhecimento.

[88] Compreendemos a completude de uma atividade como a finalização de 100% de suas tarefas.

[89] De acordo com nosso raciocínio, a não completude da atividade corresponderia ao número 0 (zero).

$$\text{Esforço da Atividade} = \frac{\text{duração da atividade}}{\text{completude da atividade}}$$

O EA pode ser interpretado, simplesmente, como a medida do tempo gasto pelo pesquisador na realização de apenas uma atividade realizada em um projeto de construção de *corpora*. Uma vez que a métrica consiste em uma forma objetiva, quantitativa e sistemática de obtermos o cálculo do EA e que se baseia em critérios e regras bem definidas, acreditamos que ela possa ser reproduzida e aplicada em diferentes projetos de construção de *corpora* e servir para que comparações entre as atividades sejam efetuadas.

No caso da nossa pesquisa, o EA foi útil para calcularmos o esforço empreendido no desenvolvimento de cada atividade realizada na elaboração do CoCLI – nas versões com e sem intervenção do *ToGatherUp*. Além disso, tomamos o EA como base para o cálculo do Esforço Total do Projeto (ETP) – métrica que criamos e utilizamos como parâmetro de comparação entre os esforços empregados na construção de cada versão do CoCLI.

O ETP é uma métrica que quantifica, em segundos, o esforço despendido para a completude[90] de um projeto de construção de um *corpus*. O modelo que definimos para determinar o cálculo do ETP é igual à soma de todos os EAs. Assim, o ETP pode ser expresso da seguinte maneira: ETP = EA 1 + EA 2 + EA 3 + EA 4 + EA *n*.

Procedemos à comparação entre as versões do CoCLI com e sem intervenção do *ToGatherUp* no que diz respeito aos esforços empregados com base no cálculo do ETP referente a cada uma das versões. Para tanto, definimos um método para o cálculo do ETP de cada uma das versões.

O método para o cálculo do ETP referente à construção da versão do CoCLI que não passou pela intervenção do *ToGatherUp* consistiu nos passos a seguir:

1. Identificamos as atividades realizadas para a construção do projeto de acordo com o *framework* de organização de projetos[91] de construção de *corpora* apresentado no tópico 2.3 A construção de um *corpus*, no

[90] Compreendemos a completude de um projeto como a finalização de 100% das suas atividades.
[91] Projetos compostos por fases, processos, atividades e tarefas.

segundo capítulo. As atividades identificadas foram: a) localização dos dados; b) permissão de uso dos dados; c) captura dos dados; d) conversão dos dados; e) limpeza e normalização dos dados; f) salvamento de arquivos; g) enriquecimento dos dados; h) nomeação dos arquivos;

2. Calculamos o EA das atividades identificadas. Para melhor compreensão, atribuímos uma sigla para cada EA calculado. Desse modo, obtivemos a seguinte lista: a) Esforço da Atividade da localização dos dados (EALD); b) Esforço da Atividade da obtenção de permissão de uso dos dados (EAOPD); c) Esforço da Atividade de captura dos dados (EACD); d) Esforço da Atividade de conversão dos dados[92] (EACVD); e) Esforço da Atividade de limpeza e normalização dos dados (EALND); f) Esforço da Atividade de salvamento de arquivos (EASA); g) Esforço da Atividade de enriquecimento dos dados (EAED); h) Esforço da Atividade de nomeação dos arquivos (EANA).
3. Aplicamos o modelo de cálculo do ETP, expresso por: ETP1[93] = EALD + EAOPD + EACD + EACVD + EALND + EASA + EAED + EANA.

No que tange ao método para o cálculo do ETP concernente à elaboração da versão do CoCLI que contou com a incorporação do *ToGatherUp*, o desenvolvemos de acordo com as etapas:

1. Identificamos as atividades realizadas para a construção do projeto de acordo com o *framework* de organização de projetos de construção de *corpora* apresentado no tópico 2.3 A construção de um *corpus* no segundo capítulo e em consonância com os recursos do *ToGatherUp*, apresentados mais adiante. As atividades identificadas[94] foram: a) localização dos dados; b) permissão de uso dos dados; c) captura dos

[92] O EACVD dos textos convertidos do formato PDF para o formato TXT é igual à soma das durações das conversões intermediárias do arquivo em formato PDF para DOC (conversão 1) e do DOC para TXT (conversão 2) que são detalhadas mais adiante.
[93] O ETP1 diz respeito ao projeto não intervencionado pelo *ToGatherUp*.
[94] As atividades a, b, c e d são comuns aos dois projetos. As atividades de salvamento, nomeação de arquivos e enriquecimento dos dados foram automatizadas pelos recursos do *ToGatherUp* e, por isso, não geraram seus respectivos EAs. Portanto, não as incluímos no cálculo do projeto intervencionado pelo *ToGatherUp*.

dados; d) conversão dos dados; e) limpeza e normalização dos dados; f) cadastramento de textos[95];

2. Calculamos o EA das atividades identificadas. De forma análoga ao passo dois do método citado anteriormente, atribuímos siglas para cada EA calculado. Logo, obtivemos a seguinte lista: a) EALD; b) EAOPD; c) EACD; d) EACVD; e) EALND; f) Esforço da Atividade de cadastramento de textos (EACT).
3. Aplicamos o modelo de cálculo do ETP, expresso por: ETP2[96] = EALD + EAOPD + EACD + EACVD + EALND + EACT.

Em ambos os projetos, fizemos a medição da duração das atividades, necessária para a obtenção do EA de cada uma das atividades, com o uso de um cronômetro disponível na interface do *ToGatherUp*. Para a obtenção da quantidade de segundos relativa à duração de uma atividade, o cronômetro foi acionado assim que a atividade foi iniciada e paralisado logo após a conclusão dela. Na sequência, procedemos com a tabulação manual da informação[97] fornecida pelo cronômetro (Instrumento 1) em uma planilha do *Google* (Instrumento 2), que serviu para a extração do conjunto de dados (*dataset*) analisado no experimento da pesquisa.

Além do EA e do ETP, estabelecemos o Esforço Total de Coleta do Texto (ETCT). O ETCT corresponde à soma de todos os EAs realizados para a inclusão de um texto em um *corpus*. O ETCT pode ser expresso de forma semelhante ao ETP. Vejamos: ETCT = EA 1 + EA 2 + EA 3 + EA 4 + EA *n.* Porém, o contexto de aplicação da expressão é limitado somente aos EAs de uma única unidade textual.

3.2 O *ToGatherUp*

Nesta seção, apresentamos, de maneira detalhada, o *ToGatherUp* – pivô desta pesquisa.

[95] O cadastramento de texto é uma atividade específica da construção de *corpora* no *ToGatherUp* e a apresentamos neste capítulo, na seção 3.2.2.2 Cadastro de textos.
[96] O ETP2 alude ao projeto intervencionado pelo *ToGatherUp*.
[97] O tempo decorrido entre o início e o fim da atividade. Ou seja, a duração da atividade.

3.2.1 ToGatherUp: o que é, para que serve e como foi feito?

O *ToGatherUp* é uma ferramenta *on-line* (www.togatherup.ileel.ufu.br) que desenvolvemos tendo em vista os princípios teóricos e metodológicos apresentados no segundo capítulo e as funcionalidades de suporte à construção manual de *corpora* do Quadro 2 – Funcionalidades de suporte à construção manual de *corpora* presentes nas ferramentas da LC. O propósito do *ToGatherUp* é oferecer suporte aos projetos de construção manual de *corpora*.

A ferramenta auxilia o pesquisador na: a) organização e arquivamento de textos; b) visualização e acompanhamento de informações sobre o *corpus* (quantidade de textos e palavras) e c) preparação dos arquivos para o uso em outras ferramentas computacionais. Ao realizarmos o carregamento (*upload*) de um texto no *ToGatherUp*, o sistema realiza, automaticamente, a inserção de cabeçalho de metadados no texto, a nomeação de acordo com uma convenção preestabelecida pelo pesquisador e o armazenamento do texto no diretório correspondente ao seu posicionamento na estrutura hierárquica do projeto. Além dessas funcionalidades, o *ToGatherUp* exibe ao pesquisador uma interface em que é possível visualizar estatísticas sobre a quantidade de textos e palavras coletada, conferindo a ele maior controle em relação ao andamento de um projeto.

A interface do *software* pode ser acessada através de um navegador de Internet e a criamos para ser amigável, simples e intuitiva, o que facilita o uso do *ToGatherUp* e proporciona economia de tempo e recursos. Para o desenvolvimento da ferramenta, lançamos mão das linguagens de programação *Hypertext Preprocessor* (PHP)[98] e *JavaScript*[99] e do banco de dados *My Structured Query Language* (MySQL)[100]. As Figuras 10 e 11 mostram a tela inicial e o logotipo do *ToGatherUp*.

[98] O PHP é uma linguagem de programação com código-fonte aberto e popular no desenvolvimento de aplicações *web*.
[99] *JavaScript* é uma linguagem de programação orientada a objeto, utilizada para controlar o código HTML e *Cascading Style Sheets* (CSS) e o comportamento de uma páginas *web*.
[100] O MySQL é um Sistema Gerenciador de Bancos de Dados (SGBD) com código-fonte aberto que faz uso da linguagem *Structured Query Language* (SQL) em sua interface de consulta.

Figura 10 – Tela de acesso do *ToGatherUp*

Fonte: *ToGatherUp*.

Figura 11 – Logotipo do *ToGatherUp*

Fonte: o autor.

Nomeamos a ferramenta como *ToGatherUp* por associação ao verbo frasal *gather up*, da língua inglesa, que, de acordo com o *Macmillan Dictionary*[101], significa "pegar coisas de lugares diferentes e colocá-las juntas" (MACMILLAN DICTIONARY, 2018)[102] . No *design* da identidade do *software*, incluímos o símbolo **輯**, um ideograma da língua japonesa que, conforme Jisho[103], um dicionário japonês *on-line*, pode ser traduzido para as seguintes palavras da língua inglesa: a) *gather*; b) *collect;* c) *compile*.

[101] Disponível em: https://www.macmillandictionary.com/dictionary/british/gather-up. Acesso em: 21 jun. 2018.
[102] Original: "*to pick up things from several different places and put them together*".
[103] Disponível em: http://jisho.org. Acesso em: 21 jun. 2018.

3.2.2 Recursos do ToGatherUp

O *ToGatherUp* possui os seguintes recursos:

1. Painel de Controle (*Data Overview*): permite a visualização da quantidade total de palavras e de textos de um *corpus*, do ETP, das quantidades de palavras e textos para cada um dos gêneros[104], tipos textuais, meios de distribuição, áreas e subáreas de um *corpus* e facilita o acompanhamento visual da evolução da coleta de textos de um *corpus*;
2. Cadastro de Textos (*Data Entry*): apresenta um formulário com os campos para a entrada dos dados do texto. Os campos são: Subárea, Título, Língua, Fonte, Gênero textual, Tipos textuais, Meio de Distribuição e ETCT;
3. Gerenciador de Textos (*Data Manager*): interface que exibe uma lista com os textos de um *corpus*, em forma de tabela, e possibilita a localização (pesquisa) de um texto (ou textos) a partir dos metadados dele;
4. Árvore de Domínio (*Domain Tree*): interface para a visualização da organização hierárquica adotada no projeto de um *corpus*;
5. Exportação de *Corpus* (*Data Exporter*): funcionalidade que exporta os arquivos de um *corpus* em diretórios organizados de acordo com a hierarquia do projeto[105].

Na sequência, abordamos cada um dos referidos recursos do *ToGatherUp*, ilustrando-os com exemplos extraídos do projeto de construção do CoCLI em que ocorreu a incorporação do *ToGatherUp* – conforme a seção 3.3 O CoCLI.

[104] Distinguimos gêneros de tipos textuais no *ToGatherUp* com base na proposta no Projeto Lácio-web. De acordo com Aluísio e Almeida (2006), o Projeto Lácio-web classifica seus textos em um gênero conforme a intenção comunicativa e o caráter discursivo deles e em um tipo textual pelo modo específico da estruturação dos textos.

[105] A exportação em diretórios permite que os arquivos sejam visualizados e manipulados por ferramentas genéricas como o *Windows Explorer* do sistema operacional *Windows*.

3.2.2.1 Painel de Controle

O Painel de Controle é a interface principal do *ToGatherUp* e exibe as informações gerais do projeto de construção de um *corpus*. O objetivo dele é oferecer ao pesquisador uma visão geral da evolução da coleta de textos do *corpus*. A Figura 12 ilustra parcialmente o Painel de Controle do *ToGatherUp* com informações do projeto do CoCLI.

Figura 12 – Painel de Controle com informações do CoCLI

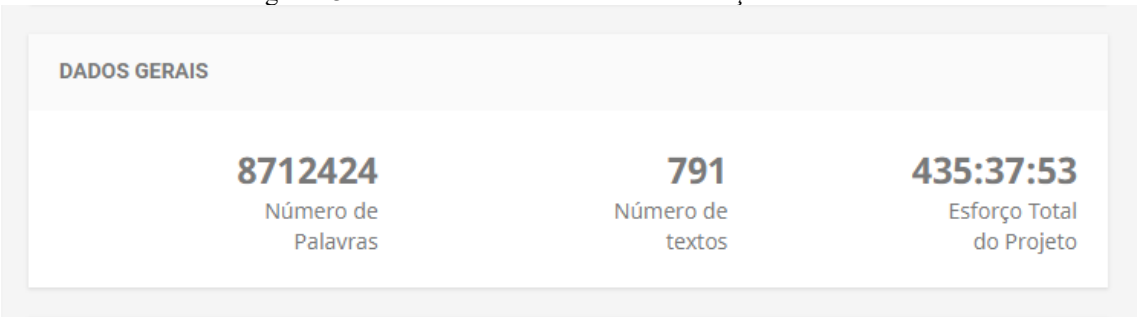
Fonte: *ToGatherUp*.

O Painel de Controle contém outros cinco painéis: Dados gerais, Textos por Gêneros, Textos por Tipos Textuais, Textos por Meios de Distribuição e Textos por Áreas e Subáreas. Descrevemos cada um deles, respectivamente, nas próximas seções. Também lançamos mão do CoCLI para mostrarmos o funcionamento deles.

3.2.2.1.1 Painel Dados Gerais

A Figura 13 alude ao painel Dados Gerais, que apresenta o número total de palavras, a quantidade total de textos e o ETP de um *corpus*.

Figura 13 – Painel Dados Gerais com informações do CoCLI

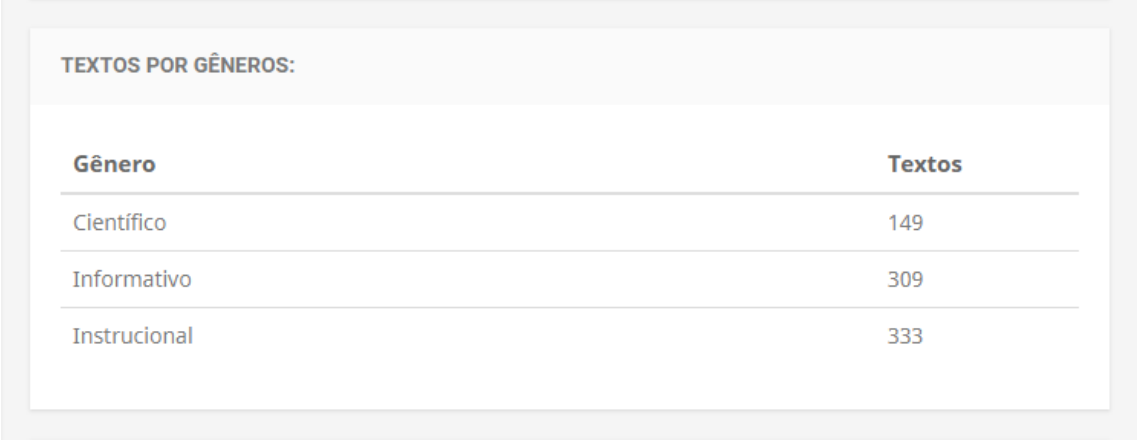


Fonte: *ToGatherUp*.

3.2.2.1.2 Painel Textos por Gêneros

A Figura 14 diz respeito ao painel Textos por Gêneros, que apresenta as quantidades totais de textos para cada gênero textual que compõe um *corpus*.

Figura 14 – Painel Textos por Gêneros com informações do CoCLI



Fonte: *ToGatherUp*.

3.2.2.1.3 Painel Textos por Tipos Textuais

A Figura 15 ilustra o painel Textos por Tipos Textuais, que apresenta o número total de textos para cada tipo textual de um *corpus*.

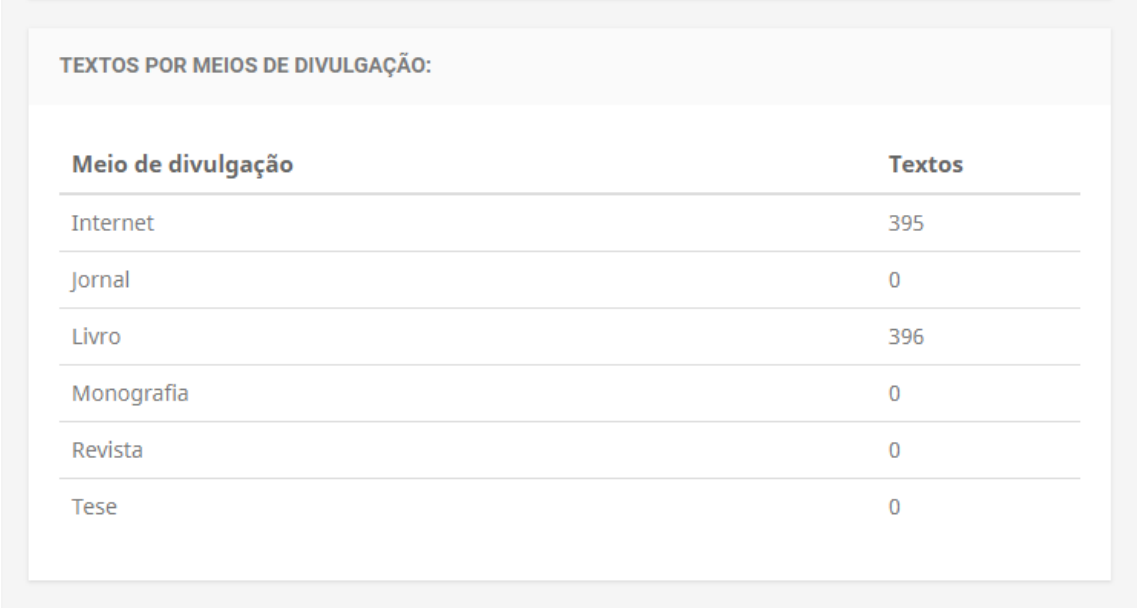
Figura 15 – Painel Textos por Tipos Textuais com informações do CoCLI

Fonte: *ToGatherUp*.

3.2.2.1.4 Painel Textos por Meios de Divulgação

A Figura 16 mostra o painel Textos por Meios de Divulgação, que apresenta, de maneira discriminada, os meios de comunicação em que os textos de um *corpus* foram obtidos durante sua coleta, quantificando-os.

Figura 16 – Painel Textos por Meios de Divulgação com informações do CoCLI

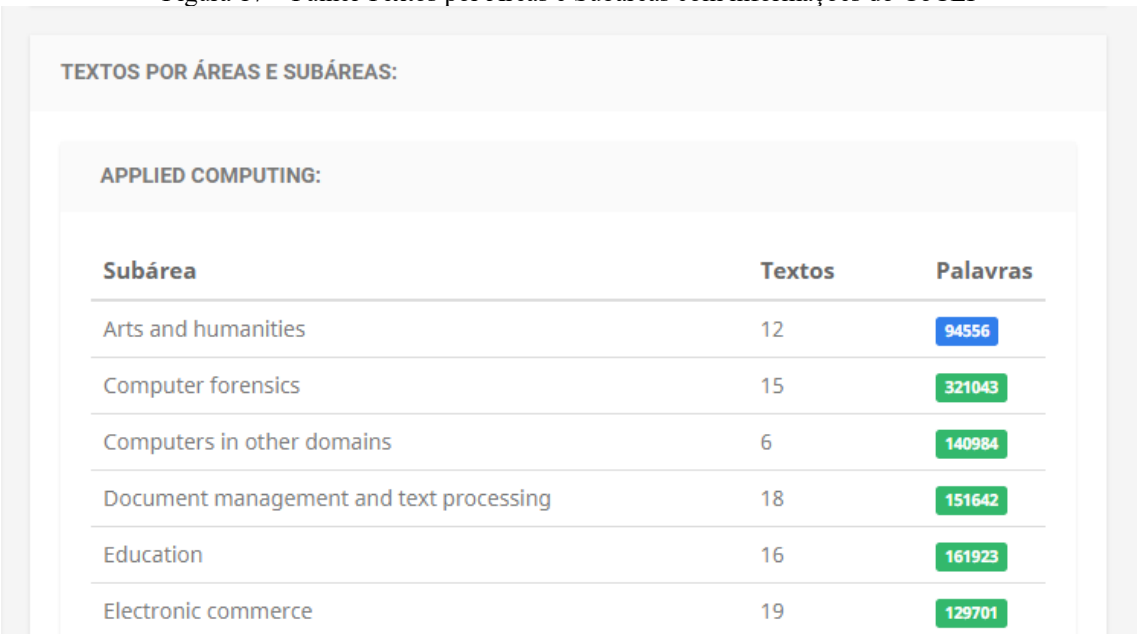


Fonte: *ToGatherUp*.

3.2.2.1.5 Painel Textos por Áreas e Subáreas

A Figura 17 exibe o Painel Textos por Áreas e Subáreas, que fornece a visão da quantidade de textos e de palavras para cada item da hierarquia adotada num projeto de construção de *corpus*. No caso do CoCLI, a hierarquia obedece à Árvore de Domínio da Computação – apresentada mais adiante no tópico 3.3.2 Árvore de Domínio da Computação do terceiro capítulo.

Figura 17 – Painel Textos por Áreas e Subáreas com informações do CoCLI



Fonte: *ToGatherUp*.

Na Figura 17, observamos informações da área *Applied Computing* e de parte de suas subáreas. A coluna "Palavras" destaca-se visualmente por apresentar números em caixas verdes e azuis que indicam ao pesquisador se a quantidade mínima de palavras[106] já foi atingida ou não. Os números em caixas verdes mostram que o mínimo de palavras esperado para a área foi alcançado e os números em caixas azuis evidenciam que ainda é necessário coletar dados para a área.

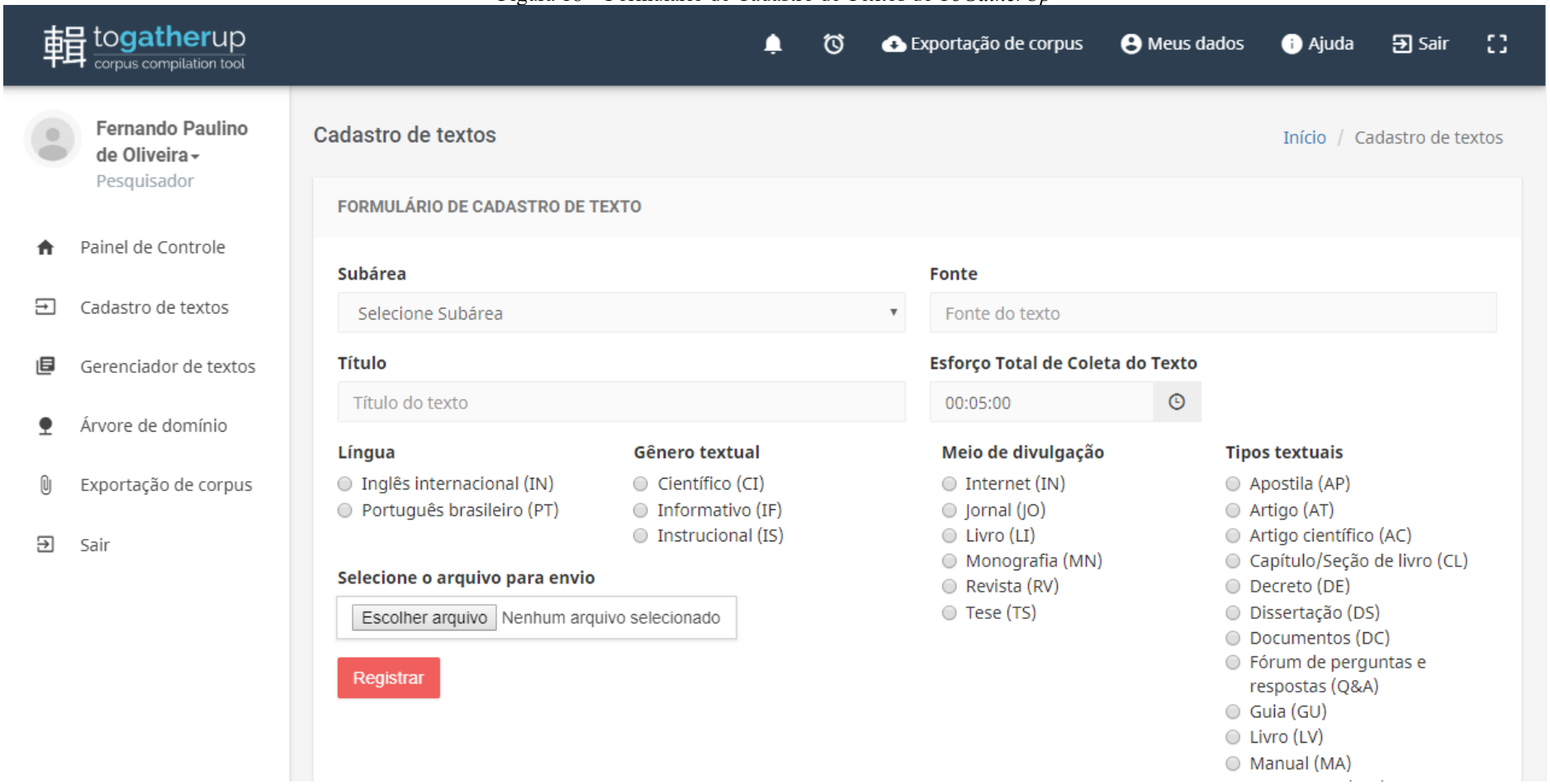
[106] A quantidade mínima de palavras é informada pelo pesquisador no momento da configuração do projeto no *ToGatherUp*.

3.2.2.2 Cadastro de Textos

No *ToGatherUp*, a inclusão de um texto em um *corpus* é realizada através do recurso denominado Cadastro de Textos, presente na Figura 18. O Cadastro de Textos é uma interface *web* que apresenta um formulário composto pelos seguintes campos[107]: (a) Subárea; (b) Título; (c) Língua; (d) Fonte; (e) Gênero Textual; (f) Tipos Textuais; (g) Meio de Distribuição; (h) ETCT. Além desses campos, o formulário apresenta, ainda, a opção para que o pesquisador possa anexar o arquivo do texto[108].

[107] Os campos citados estão de acordo com os critérios que estabelecemos para o projeto do CoCLI (detalhados mais adiante nesta Dissertação) e podem ser configurados de forma diferente para projetos que venham a adotar a ferramenta. Os campos do Cadastro de Textos podem ser definidos pelo pesquisador no momento da configuração do projeto no *ToGatherUp*. A data de publicação do texto e a sua autoria são exemplos de informações que podem ser incluídas durante a configuração do projeto.
[108] O *ToGatherUp* aceita somente arquivos no formato TXT.

Figura 18 – Formulário de Cadastro de Textos do *ToGatherUp*



Fonte: *ToGatherUp*.

Ao clicarmos na opção “Registrar” do formulário do Cadastro de Textos, uma série de atividades é executada, de forma automática, pelo *ToGatherUp*:

- Atividade 1: Registro dos metadados do texto no banco de dados;
- Atividade 2: Nomeação[109] do arquivo do texto;
- Atividade 3: Inserção de cabeçalho no arquivo do texto;
- Atividade 4: Armazenamento do arquivo do texto.

Na sequência, descrevemos cada uma dessas atividades e como elas foram aplicadas na nossa pesquisa.

3.2.2.2.1 Atividade 1: Registro dos metadados do texto no banco de dados

Ao armazenar os textos de um *corpus*, o pesquisador precisa estabelecer padrões descritivos que otimizem o acesso a eles, a recuperação e o reuso deles. Diante dessa necessidade, adotamos no *ToGatherUp* o uso de metadados para a catalogação dos textos dos nossos *corpora*. A utilização de metadados surgiu no âmbito das Ciências da Informação como uma solução para a organização de dados. Para Alves (2010), os metadados podem ser definidos como:

> [...] atributos que representam uma entidade (objeto do mundo real) em um sistema de informação. Em outras palavras, são elementos descritivos ou atributos referenciais codificados que representam características próprias ou atribuídas às entidades; são ainda dados que descrevem outros dados em um sistema de informação, com o intuito de identificar de forma única uma entidade (recurso informacional) para posterior recuperação (ALVES, 2010, p. 47).

Fizemos a definição dos campos do formulário de Cadastro de Textos de acordo com os critérios do desenho do CoCLI. Utilizamos as informações de cada um dos campos do formulário para a criação dos metadados dos textos. Desse modo, os metadados do CoCLI apresentam-se conforme o Quadro 4.

[109] Na realidade, o que ocorre é uma renomeação, porque, para que seja possível a sua submissão no *ToGatherUp*, o arquivo precisa ter sido previamente salvo pelo pesquisador. O *ToGatherUp* desconsidera qualquer que seja o nome dado a um arquivo submetido a ele e procede com a sua renomeação em conformidade com os metadados do texto e com a convenção de nomeação de arquivos do projeto.

Quadro 4 – Metadados do CoCLI

Metadados	Descrição
(a) Subárea	Informa a subárea do texto.
(b) Título	Informa o nome dado para o texto.
(c) Língua	Informa o idioma em que o texto foi escrito.
(d) Fonte	Informa a origem do texto.
(e) Gênero textual	Informa o gênero textual do texto.
(f) Tipos textuais	Informa o tipo textual do texto.
(g) Meio de distribuição	Informa o meio em que o texto foi divulgado.
(h) ETCT	Informa o esforço total referente à soma de todos os EAs realizados para a inclusão de uma unidade textual no *corpus*[110].

Fonte: o autor.

Além dos metadados do Quadro 4, o Quadro 5 apresenta um conjunto de metadados que o *ToGatherUp* registra, de forma automática, sem que ocorra a intervenção do pesquisador.

Quadro 5 – Metadados gerados de forma automática pelo *ToGatherUp*

Metadados	Descrição
(i) Domínio	Informa o domínio do texto (área do conhecimento/especialidade a qual pertence)[111].
(j) Número de palavras	Informa o número de palavras do texto[112].
(k) Data da inclusão	Informa a data e a hora em que o texto foi incluído no *corpus*[113].
(l) Identificador do arquivo (ID)	Informa o número de identificação do texto no banco de dados do *ToGatherUp*[114].

Fonte: dados do autor.

[110] A obtenção do ETCT depende do registro do EA de cada uma das atividades necessárias para a coleta do texto. É importante lembrar que o *ToGatherUp* não apresenta uma forma de registro para cada um dos EAs. O *software* tem somente um cronômetro que pode ser utilizado para a captura da duração de cada atividade, que pode ser registrada em um tipo de controle escolhido pelo pesquisador.
[111] O domínio do texto é estabelecido durante as configurações do projeto no *ToGatherUp*. Por essa razão, o *ToGatherUp* é capaz de incluí-lo, automaticamente, como um metadado.
[112] O *ToGatherUp* possui um algoritmo que contabiliza a quantidade de palavras do texto.
[113] O *ToGatherUp* considera a data e a hora do servidor em que o sistema está instalado. Por isso, o pesquisador não precisa informar esses dados.
[114] O ID é gerado de forma incremental e automática pelo MySQL.

Ao preenchermos os campos do formulário do Cadastro de Textos e realizarmos a submissão do texto, o *ToGatherUp* efetuou o registro dos metadados descritos nos Quadros 4 e 5 no banco de dados do sistema.

3.2.2.2.2 Atividade 2: Nomeação dos arquivos dos textos

O *ToGatherUp* faz a nomeação automática dos textos de um *corpus* durante a submissão deles pelo Cadastro de Textos de acordo com os metadados do texto e com uma convenção de nomeação de arquivos definida durante a configuração do projeto no sistema. Com base na convenção que estabelecemos para a nomeação dos textos do CoCLI, um dos textos do *corpus* foi nomeado, por exemplo, desta forma: IN-CO-IF-AT-IN-25Sep2017-797.txt.

O nome do arquivo é constituído por sete partes distintas, separadas por hífen, e finalizado com a extensão correspondente ao formato dele (.txt). Cada uma das partes é formada por uma abreviação que se associa a um metadado do texto:

a) a primeira (IN) informa a língua do texto. Para a língua inglesa, utilizamos a abreviação IN;
b) a segunda (CO) diz respeito ao domínio (área do conhecimento/especialidade a que pertence o texto). Como o CoCLI é do domínio da Computação, utilizamos CO para abreviá-lo;
c) a terceira (IF) refere-se ao gênero do texto. Usamos a abreviação CI para o gênero científico, a IF para o gênero informativo e a IS para o gênero instrucional;
d) a quarta (AT) alude ao tipo do texto. Estabelecemos as abreviações referentes aos tipos textuais dos textos do CoCLI da seguinte maneira:

- Apostila (AP);
- Artigo (AT);
- Artigo científico (AC);
- Capítulo/Seção de livro (CL);
- Decreto (DE);
- Dissertação (DS);

- Documentos (DC);
- Fórum de perguntas e respostas (Q&A);
- Guia (GU);
- Livro (LV);
- Manual (MA);
- Monografia (MN);
- Norma técnica (NR);
- Nota técnica (NT);
- Notícia (NO);
- Portaria (PA);
- Relatório (RL);
- Reportagem (RP);
- Tese (TS);
- Transcrição (TR);
- Tutorial (TT).

e) a quinta parte (IN) é relativa ao meio de divulgação do texto. Como todos os textos dos nossos *corpora* são provenientes da Internet, utilizamos a sigla IN para representá-la;
f) a sexta parte (25Sep2017) informa a data de coleta do texto;
g) a sétima parte (797) indica o identificador (ID) do texto no banco de dados do *ToGatherUp*. Cada texto recebe um ID único ao ser registrado no banco de dados do sistema, o que evita a possibilidade de que textos com metadados idênticos recebam um mesmo nome.

Após o esclarecimento das regras e dos metadados que foram utilizados para a criação da convenção de nomeação dos arquivos do CoCLI que foi configurada no *ToGatherUp*, podemos dizer que o arquivo citado como exemplo anteriormente (IN-CO-IF-AT-IN-25Sep2017-797.txt.) trata de um artigo escrito em língua inglesa, pertencente ao domínio da Computação, de gênero informativo, retirado da Internet e inserido no *corpus* em 25 de setembro de 2017 sob o identificador 797.

3.2.2.2.3 Atividade 3: Inserção de cabeçalho nos arquivos de texto

Os metadados dos textos do CoCLI foram usados pelo *ToGatherUp* para a criação e inserção automática de cabeçalho nos arquivos dos textos. Para que isso fosse possível, informamos nas configurações da ferramenta a estrutura do cabeçalho a ser utilizada. Considerando que realizamos o projeto de construção do CoCLI, exclusivamente, para a avaliação dos efeitos da incorporação do *ToGatherUp* na construção manual de *corpus*, decidimos que a estrutura do cabeçalho dos textos deveria conter apenas o mínimo de informação: a origem do texto e a sua data de inclusão no *corpus*. Com base nisso, o *ToGatherUp* procedeu com a inserção do cabeçalho nos textos, alimentando-os com os metadados fornecidos no Cadastro de Textos da ferramenta. Dessa maneira, um dos textos do CoCLI recebeu o cabeçalho apresentado na Figura 19.

Figura 19 – Exemplo de cabeçalho de um texto do CoCLI

```
<textHeader>
  <sourceText>
    <pubPlace> http://www.informs-sim.org/wsc08papers/007.pdf </pubPlace>
    <accessDate> 2017-09-01 08:20:39 </accessDate>
  </sourceText>
</textHeader>
```

Fonte: *ToGatherUp*.

A estrutura do cabeçalho segue o padrão XML e apresenta as etiquetas <*textHeader*>, <*sourceText*>, <*pubPlace*> e <*accessDate*>, organizadas assim: a etiqueta <*textHeader*> ocupa o nível primário na hierarquia do cabeçalho e delimita o início e o fim dele[115]. A etiqueta <*sourceText*>, de nível secundário, é aninhada à <*textHeader*> por estabelecer uma relação de parentesco com ela. A função da <*sourceText*> é a de agrupar as demais etiquetas, <*pubPlace*> e <*accessDate*>, ambas de nível terciário. A etiqueta <*pubPlace*> indica a origem do texto e a <*accessDate*> marca a data de inclusão do texto no *corpus*.

[115] A delimitação do início e do fim do cabeçalho por etiquetas é importante, já que, quando o pesquisador as insere adequadamente, as informações dele são ignoradas pelas ferramentas de análise de texto.

3.2.2.2.4 Procedimento 4: Armazenamento do arquivo do texto

O armazenamento de arquivos através de métodos tradicionais comuns no cotidiano das organizações e no gerenciamento de informações pessoais é natural. No entanto, Dourish (2003, p. 4) aponta que estudos realizados por Barreau e Nardi (1995) e por Kaptelinin (1996) revelam que essa prática é problemática, pois dificulta a reorganização das informações quando elas assumem funções diferentes das originais ou quando elas não se adequam a somente um dos *loci* de armazenamento.

Considerando essa problemática, desenvolvemos o *ToGatherUp* de modo que ele fosse capaz tanto de armazenar os textos do CoCLI de acordo com a Árvore de Domínio da Computação (uma estrutura hierárquica fixa) como de reorganizá-los seguindo outras configurações hierárquicas. Para alcançarmos esse objetivo, incorporamos ao *ToGatherUp* um modelo conceitual de gerenciamento de arquivos chamado *Placeless Documents.* O *Placeless Documents* foi criado por Paul Dourish (2003), pesquisador do *Xerox Palo Alto Research Center*, localizado em Palo Alto, na Califórnia, nos Estados Unidos, e propõe a organização de documentos a partir das suas propriedades, conforme as diferentes necessidades de seus usuários.

Nesse modelo, a associação das propriedades dos documentos (informações sobre os próprios documentos), chamadas de *active properties*, aos documentos permite que eles sejam organizados de acordo com essas propriedades ao invés de obedecerem a uma estrutura hierárquica predeterminada. Ao oferecer essa nova forma de organização baseada em propriedades, o *Placeless Documents* possibilita o agrupamento, de diferentes maneiras, de um conjunto de documentos, o que soluciona o problema da reorganização de arquivos de acordo com suas funções. A flexibilidade proporcionada pelo *Placeless Documents* foi a principal razão que nos levou a incorporá-lo ao *ToGatherUp*. No entanto, o modelo que acrescentamos ao *ToGatherUp* baseou-se na associação dos metadados dos textos do CoCLI ao invés das propriedades dos seus arquivos.

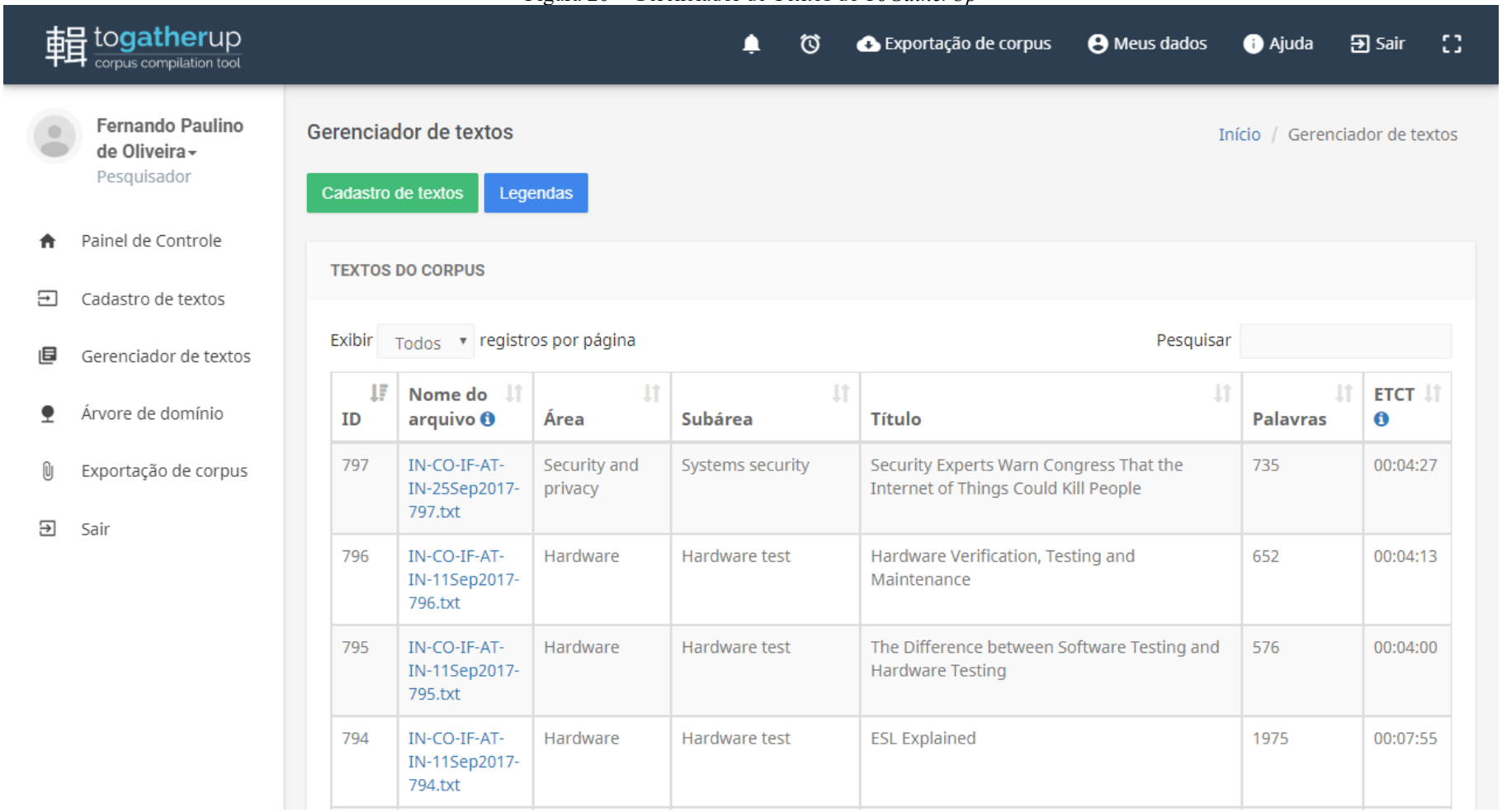
Ao submetermos um texto por meio do formulário do Cadastro de Textos do *ToGatherUp*, seus metadados são registrados no banco de dados do sistema e seu arquivo é armazenado em um diretório comum do servidor *web* em que o sistema está instalado. Por seguir o modelo *Placeless Documents*, o local de armazenamento dos arquivos dentro da infraestrutura do *ToGatherUp* é irrelevante, uma vez que será a necessidade do pesquisador que irá determinar seu posicionamento na estrutura de

diretórios, que será gerada no momento da sua exportação para o processamento em outras ferramentas computacionais.

3.2.2.3 Gerenciador de Textos

Além das informações quantitativas disponíveis no Painel de Controle, o *ToGatherUp* apresenta uma interface, nomeada como Gerenciador de Textos, que permite ao pesquisador a visualização dos textos de um *corpus*, em forma de tabela, e a pesquisa por um ou mais textos do *corpus* com base em suas informações. A Figura 20 mostra a interface do Gerenciador de Textos.

Figura 20 – Gerenciador de Textos do *ToGatherUp*



Fonte: *ToGatherUp*.

A tabela do Gerenciador de Textos possui as colunas: a) ID; b) Nome do arquivo; c) Área; d) Subárea; e) Título; f) Palavras (número de palavras); g) ETCT. O clique sobre o título de cada coluna faz com que suas informações sejam visualizadas em ordem crescente ou decrescente, no caso dos dados numéricos, ou em ordem alfabética, no caso dos dados alfabéticos ou alfanuméricos. O clique sobre o nome do arquivo faz com que o seu conteúdo seja exibido no navegador de Internet.

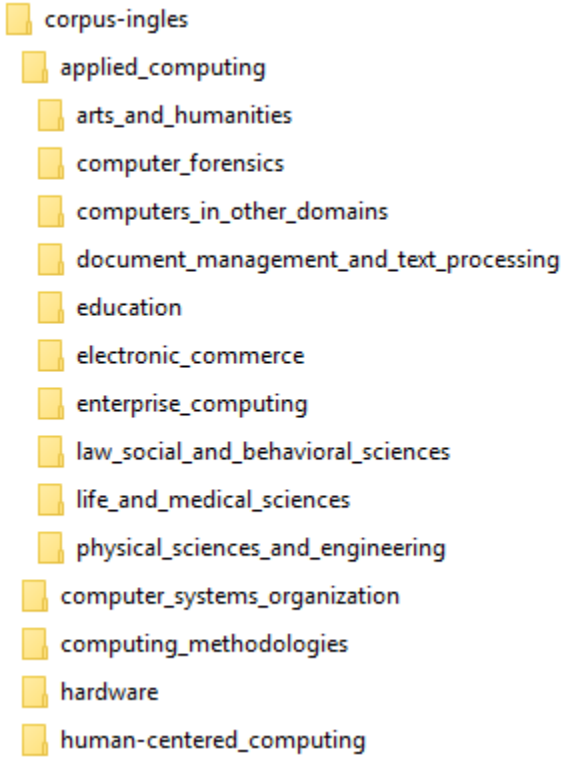
3.2.2.4 Exportação de Corpus

Ao término do projeto de construção de um *corpus*, os seus dados precisam ser disponibilizados para o processamento em ferramentas computacionais. Pensando nisso, o *ToGatherUp* possui o recurso Exportação de *Corpus*, que operacionaliza a exportação dos arquivos do *corpus* de modo que eles possam ser manipulados por ferramentas convencionais de gerenciamento de arquivos, como o *Windows Explorer*, do sistema operacional *Windows*, e importados nas ferramentas que irão processá-los.

Para realizar a exportação, o pesquisador deve utilizar a opção Exportação de *Corpus*, disponível na barra superior e no menu principal do sistema, a qualquer momento que julgar necessário. Ao acioná-la, o *ToGatherUp* cria, de forma automática, um arquivo compactado contendo os textos do *corpus* organizados em diretórios e subdiretórios, conforme a estrutura estabelecida para o projeto, nomeados de forma padronizada e com os seus cabeçalhos já inseridos no início do conteúdo de cada arquivo. A exportação do *corpus* para uso em outras ferramentas de gerenciamento de arquivos é importante, porque amplia as possibilidades de adoção da ferramenta por usuários que precisam utilizar outras formas de visualização dos dados diferentes da oferecida pelo *ToGatherUp*.

No caso da exportação dos dados do CoCLI, o *ToGatherUp* criou, automaticamente, um arquivo compactado com a extensão .zip, contendo os textos do *corpus* alocados em diretórios correspondentes aos seus campos semânticos na Árvore de Domínios da Computação. A Figura 21 ilustra parte da estrutura de diretórios do CoCLI e a Figura 22 mostra o conteúdo do subdiretório *arts_and_humanities*, correspondente à subárea *Arts and humanities*, da área *Applied computing* do CoCLI.

Figura 21 – Parte da estrutura de diretórios do CoCLI



Fonte: o autor.

Figura 22 – Textos da subárea *Arts and humanities*, da área *Applied computing*, do CoCLI

Nome	Data de modificação	Tipo	Tamanho
IN-CO-CI-AC-IN-17Jul2017-226.txt	17/07/2017 11:44	Documento de Texto	36 KB
IN-CO-CI-AC-IN-17Jul2017-227.txt	17/07/2017 11:52	Documento de Texto	24 KB
IN-CO-CI-AC-IN-17Jul2017-229.txt	17/07/2017 12:26	Documento de Texto	23 KB
IN-CO-CI-AC-IN-17Jul2017-230.txt	17/07/2017 12:37	Documento de Texto	9 KB
IN-CO-CI-AC-IN-17Jul2017-231.txt	17/07/2017 12:42	Documento de Texto	37 KB
IN-CO-CI-AC-IN-17Jul2017-232.txt	17/07/2017 12:48	Documento de Texto	47 KB
IN-CO-CI-AC-IN-17Jul2017-233.txt	17/07/2017 13:01	Documento de Texto	44 KB
IN-CO-CI-AT-IN-07Sep2017-790.txt	07/09/2017 10:28	Documento de Texto	24 KB
IN-CO-IF-AT-IN-01Sep2017-741.txt	01/09/2017 10:45	Documento de Texto	51 KB
IN-CO-IF-AT-IN-01Sep2017-745.txt	01/09/2017 14:31	Documento de Texto	31 KB
IN-CO-IF-AT-IN-07Sep2017-789.txt	07/09/2017 10:24	Documento de Texto	15 KB
IN-CO-IS-LV-LI-01Sep2017-742.txt	01/09/2017 11:12	Documento de Texto	252 KB

Fonte: o autor.

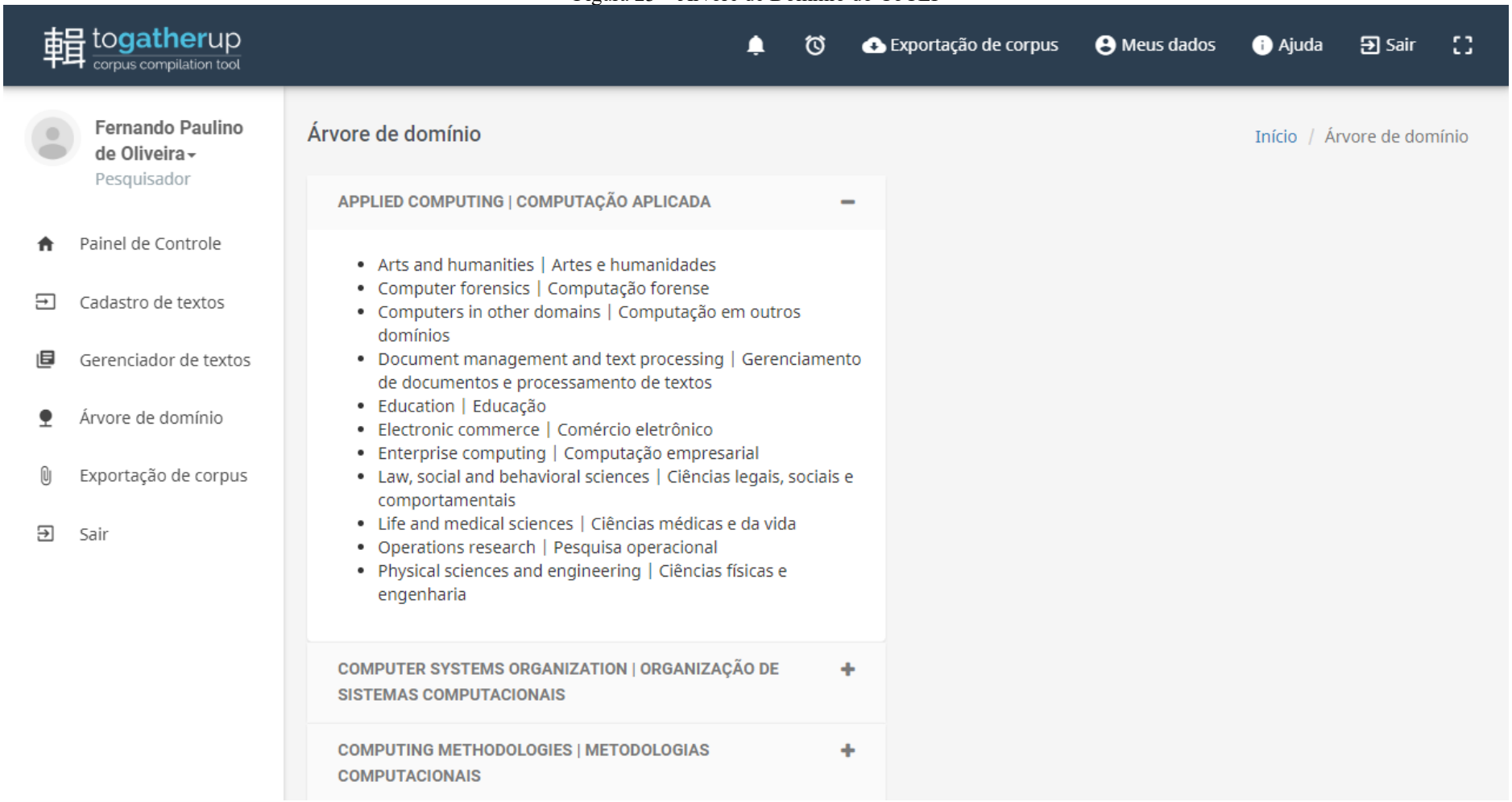
É importante salientarmos que, embora a nossa pesquisa tenha exigido que os textos do CoCLI fossem exportados conforme os campos nocionais da Árvore de Domínio da Computação, a flexibilidade oferecida pelo modelo *Placeless documents* e pelo uso dos metadados permite que a exportação seja feita de acordo com outros esquemas. Para exemplificarmos, poderíamos gerar, a partir do conjunto de textos do

CoCLI, um *subcorpus* composto somente por textos do gênero científico, caso as configurações de exportação do *ToGatherUp* fossem definidas para esse novo esquema.

3.2.2.5 Árvore de Domínio

A Árvore de Domínio é a interface do *ToGatherUp* que exibe a organização hierárquica adotada no projeto de construção de um *corpus*. A Figura 23 exibe a Árvore de Domínio da Computação, com suas áreas e subáreas, adotada no nosso projeto de construção do CoCLI.

Figura 23 – Árvore de Domínio do CoCLI



Fonte: *ToGatherUp*.

3.3 O CoCLI

Nesta seção, discorremos sobre os dois métodos que usamos para a construção do CoCLI e os instrumentos que utilizamos para a tabulação dos dados que possibilitaram a comparação entre os esforços despendidos para a execução dos projetos de cada uma das versões do CoCLI. Com o intuito de facilitar a compreensão do texto, utilizamos a expressão "Método 1" para referenciar o método que não envolveu a incorporação do *ToGatherUp* e "Método 2" para o que adotou a ferramenta. Antes disso, apresentamos brevemente a área da Computação e explicamos como escolhemos a Árvore de Domínio utilizada nos dois projetos.

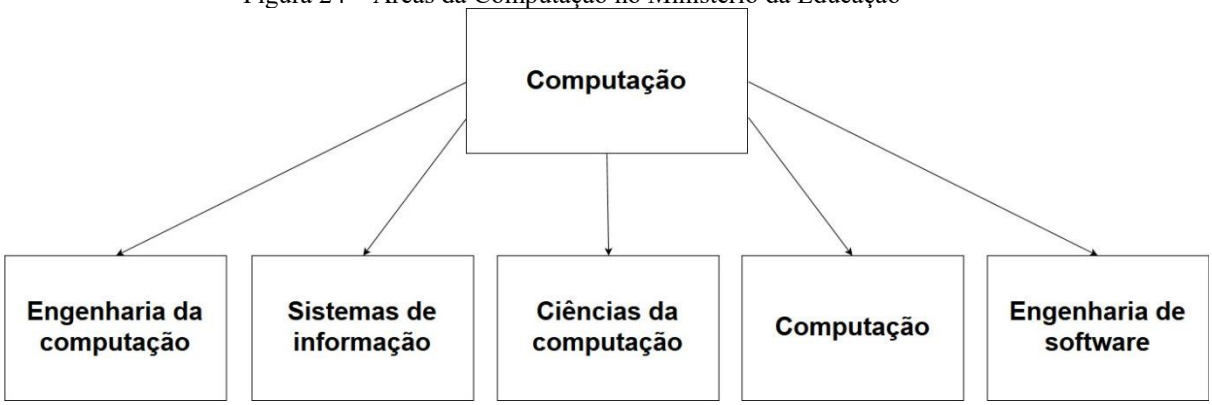
3.3.1 Apresentação da área da Computação

Ao longo da história da humanidade, o homem procurou fazer uso do seu intelecto para desenvolver ferramentas em benefício próprio. Essa busca resultou em um sistemático acúmulo de conhecimentos e no surgimento de tecnologias. Dentre as tecnologias criadas pelo homem, uma tem destaque central na sociedade moderna – a Computação.

No final da década de 1980, com a criação dos computadores pessoais, a Computação passou a exercer uma influência radical em, praticamente, todas as atividades humanas. De modo geral, podemos dizer que a sociedade associa a palavra "computação" a uma miríade de tópicos relacionados aos computadores. Nesta pesquisa, utilizamos a palavra Computação para referenciar a área de conhecimento ou de especialidade que abriga um conjunto de subáreas, como a Tecnologia da Informação, os Sistemas de Informação e as Ciências da Computação.

No Brasil, a Resolução CNE/CES nº 5, de 16 de novembro de 2016, do Conselho Nacional de Educação e da Câmara de Educação Superior, instituiu nas Diretrizes Curriculares Nacionais direcionadas aos cursos de graduação na área da Computação que esta abrange os cursos de bacharelado em Ciências da Computação, de Sistemas de Informação, de Engenharia da Computação, de Engenharia de *Software* e de Licenciatura em Computação, conforme ilustramos na Figura 24.

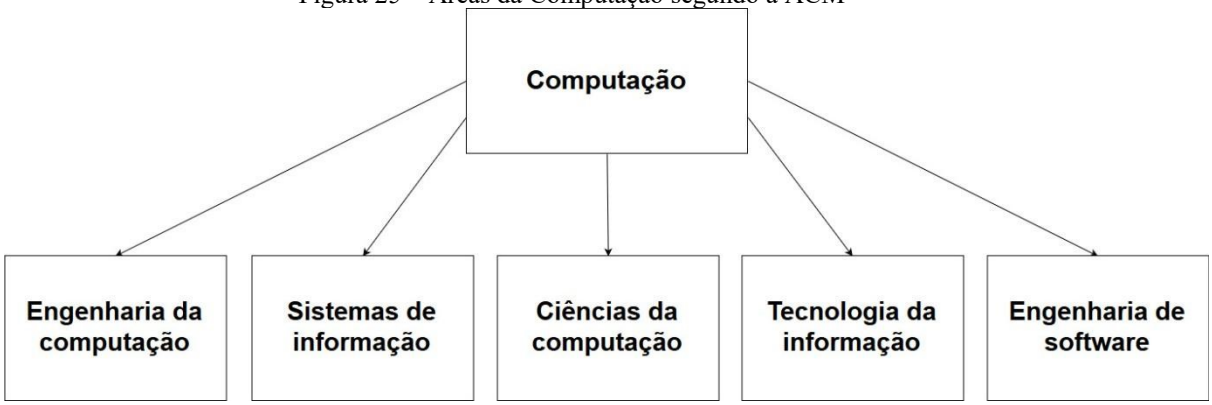
Figura 24 – Áreas da Computação no Ministério da Educação



Fonte: o autor.

No âmbito internacional, a *Association for Computing Machinery*[116], também conhecida como ACM, sigla pela qual passaremos a denominá-la, em parceria com a *Association for Information Systems* (AIS) e a *Computer Society* (CS), divide a área da Computação em cinco subáreas (cf. Figura 25): *Computer Engineering* (CE), *Computer Science* (CS), *Information Systems* (IS), *Information Technology* (IT) e *Software Engineering* (SE). A divisão foi publicada em 2005, no documento *Computing Curricula 2005*, conforme Shackelford *et al*. (2005), e serve como diretriz para a definição dos currículos dos cursos da área da Computação no mundo todo.

Figura 25 – Áreas da Computação segundo a ACM



Fonte: o autor.

Tanto a Resolução CNE/CES nº 5 quanto a AMC representam as bases contemporâneas oficiais para que as instituições de ensino possam organizar os currículos de seus cursos. No entanto, Fonseca Filho (2007) alerta para a existência de

[116] A ACM é uma organização científica e profissional dedicada à Computação desde sua fundação em 1947. É reconhecida internacionalmente pelos profissionais e cientistas da Computação e publica, desde 1968, as recomendações curriculares para os cursos das Ciências da Computação e das Ciências da Informação. O *site* da organização é: http://www.acm.org.

uma inadequação nas tentativas de definição das áreas da Computação devido à riqueza e à dinamicidade dela:

> [...] o avanço da Computação foi exponencial, abrindo-se em um grande leque de tecnologias, conceitos, ideias, transformando-se em uma figura quase irreconhecível. Atualmente falar de estado da arte na Computação tornou-se sem sentido: sob que ótica, perspectiva, campo ou área? Apesar da sua recente irrupção na história contemporânea, a partir dos anos 40 do século XX, ela já se tornou complexa, ampla, geradora de novos enfoques, tornando-se um verdadeiro desafio a quem queira entendê-la e traçar sua evolução (FONSECA FILHO, 2007, p. 23).

A inadequação apontada por Fonseca Filho (2007) repercutiu na nossa pesquisa no momento da definição da Árvore de Domínio que seria utilizada nos projetos de construção do CoCLI, conforme esclarecemos no próximo tópico.

3.3.2 Árvore de Domínio da Computação

Definimos a Árvore de Domínio que utilizamos nos projetos de construção do CoCLI durante o processo de planejamento da construção dos *corpora*, mais precisamente, no momento em que estabelecemos o desenho dos *corpora*.

No início, pretendíamos adotar a Árvore de Domínio da Computação proposta por Fromm (2002). Porém, ao ponderarmos sobre o intervalo de tempo decorrido entre a elaboração da proposta do referido autor e a realização da nossa pesquisa, entendemos que necessitaríamos atualizá-la. Para tanto, estudamos a taxonomia das Ciências da Computação, apresentada pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)[117] e por centros acadêmicos de referência[118], e constatamos a inexistência de uma classificação padronizada, o que refletiu a inadequação apontada por Fonseca Filho (2007) anteriormente.

Diante disso, decidimos formular uma nova proposta de Árvore de Domínio da Computação que contemplasse as convergências entre as classificações do nosso estudo, a proposta de Fromm (2002) e, na medida do possível, as novas áreas e subáreas

[117] Disponível em: http://lattes.cnpq.br/web/dgp/ciencias-exatas-e-da-terra. Acesso em: 21 jun. 2018.
[118] Universidade Federal do Rio de Janeiro (UFRJ), Universidade de São Paulo (USP), Universidade Estadual de Campinas (UNICAMP), Universidade Estadual de Campinas (UNICAMP), Universidade Federal do Rio Grande do Sul (UFRGS), Universidade Federal de Pernambuco (UFPE), Universidade Federal de Minas Gerais (UFMG), *Massachusetts Institute of Technology* (MIT), *Harvard University*, *Stanford University* e *University of Cambridge*.

identificadas por nós durante a leitura de materiais da área da Computação. Com esse objetivo em mente, após um árduo trabalho, criamos uma primeira proposta de Árvore de Domínio da Computação, presente no **APÊNDICE B – Primeira proposta de Árvore de Domínio da Computação**.

A primeira proposta possuía 38 áreas que se ramificavam em até cinco níveis de profundidade, totalizando 247 campos conceituais. A grande quantidade de campos conceituais, somada à definição da quantidade mínima de 100 mil palavras para cada campo conceitual que estabelecemos na intenção de garantir um melhor balanceamento dos *corpora* da pesquisa, mostrou-se problemática devido ao esforço necessário para a reunião do volume de informações para cada um dos *corpora* da pesquisa e às restrições de tempo para a completude do curso de Mestrado.

À vista dessa dificuldade, decidimos reconfigurar nossa proposta inicial, eliminando os subníveis de profundidade cinco. Desse modo, chegamos à segunda proposta de Árvore de Domínio da Computação, apresentada no **APÊNDICE C – Segunda proposta de Árvore de Domínio da Computação**. Apesar de todo o esforço para a construção das propostas citadas, tínhamos a consciência de que elas representavam apenas o "nosso recorte" da realidade. Essa percepção consistia em um grande desconforto até que nossas leituras apontaram para a existência de um sistema de classificação da computação chamado *Computing Classification System* (CCS)[119] – disponível no **ANEXO A – *Computing Classification System* (CSS)**.

O CCS, criado em 2012, é a classificação da área da Computação proposta pela ACM, em esforço conjunto com a AIS e a *Computer Society* (IEEE-CS), instituições amplamente reconhecidas pelos especialistas da área da Computação. O CCS foi desenvolvido por métodos de análise estatística de um extenso *corpus* de textos da Computação e apresenta-se na forma de uma ontologia hierárquica cujas categorias e conceitos podem ser considerados como o mais próximo de um "estado da arte" da classificação da área da Computação. Diante da descoberta e da representatividade do CSS, abandonamos a ideia de propormos uma Árvore de Domínio da Computação e decidimos adotá-lo como a Árvore de Domínio da nossa pesquisa.

O CCS é organizado em 12 grandes áreas (*Hardware*, *Computer systems organization*, *Networks*, *Software and its engineering*, *Theory of computation*, *Mathematics of computing*, *Information systems*, *Security and privacy*, *Human-centered*

[119] Disponível em: https://www.acm.org/publications/class-2012. Acesso em: 21 jun. 2018.

computing, *Computing methodologies*, *Applied computing* e *Social and professional topics*) que se dividem em subáreas que podem se ramificar em até cinco níveis de profundidade, totalizando 2074 territórios conceituais. Novamente, esbarramos na questão do volume de dados e da exequibilidade do nosso projeto e, mais uma vez, usamos a estratégia de redução de níveis para resolver o problema. Dessa vez, eliminamos os subníveis de profundidade 3, 4 e 5 do CCS. Como resultado, a proposta simplificada do CCS que adotamos passou a apresentar a configuração do Quadro 6.

Quadro 6 – CSS simplificado

Computing

Hardware

Printed circuit boards
Communication hardware, interfaces and storage
Integrated circuits
Very large scale integration design
Power and energy
Electronic design automation
Hardware validation
Hardware test
Robustness
Emerging technologies

Computer systems organization

Architectures
Embedded and cyber-physical systems
Real-time systems
Dependable and fault-tolerant systems and networks

Networks

Network architectures
Network protocols
Network components
Network algorithms
Network performance evaluation
Network properties
Network services
Network types

Software and its engineering

Software organization and properties

Software notations and tools
Software creation and management

Theory of computation
Models of computation
Formal languages and automata theory
Computational complexity and cryptography
Logic
Design and analysis of algorithms
Randomness, geometry and discrete structures
Theory and algorithms for application domains
Semantics and reasoning

Mathematics of computing
Discrete mathematics
Probability and statistics
Mathematical software
Information theory
Mathematical analysis
Continuous mathematics

Information systems
Data management systems
Information storage systems
Information systems applications
World Wide Web
Information retrieval

Security and privacy
Cryptography
Formal methods and theory of security
Security services
Intrusion/anomaly detection and
malware mitigation
Security in hardware
Systems security
Network security
Database and storage security
Software and application security
Human and societal aspects of security and privacy

Human-centered computing
Human computer interaction (HCI)

Interaction design
Collaborative and social computing
Ubiquitous and mobile computing
Visualization
Accessibility

Computing methodologies
Symbolic and algebraic manipulation
Parallel computing methodologies
Artificial intelligence
Machine learning
Modeling and simulation
Computer graphics
Distributed computing methodologies
Concurrent computing methodologies

Applied computing
Electronic commerce
Enterprise computing
Physical sciences and engineering
Life and medical sciences
Law, social and behavioral sciences
Computer forensics
Arts and humanities
Computers in other domains
Operations research
Education
Document management
and text processing

Social and professional topics
Professional topics
Technology policy
User characteristics

Fonte: o autor.

3.3.3 Os projetos de construção do CoCLI

A realização dos projetos de construção das duas versões do CoCLI seguiu os fundamentos apresentados no segundo capítulo desta Dissertação. Os dois métodos apresentam um conjunto de atividades em comum e um conjunto de atividades próprias

de cada um deles. A seguir, descrevemos a parte comum entre os métodos e, na sequência, tratamos da parte em que eles se distinguem um do outro.

3.3.3.1 Parte comum entre os métodos 1 e 2

A parte comum entre os métodos 1 e 2 compreende atividades das fases inicial e de execução dos projetos de construção de *corpora*. A primeira atividade que realizamos foi a definição do desenho do *corpus*. Após a seleção e definição dos critérios, o desenho do CoCLI apresentou a configuração do Quadro 7.

Quadro 7 – Desenho do CoCLI

Critério	Definição
Objetivo	Recuperar informações, extrair termos, definir termos e identificar exemplos de uso de termos.
Domínio[120]:	Textos restritos às áreas e subáreas da Computação.
Tipo	Especializado (composto por textos das áreas e subáreas da Computação).
Tempo[121]	Sincrônico (contempla textos publicados no período de 2000 a 2018).
Língua	Monolíngue (apenas textos escritos na língua inglesa)[122].
Gênero e tipo textual	Textos científicos (artigos científicos, capítulos/seções de livro, teses, dissertações, monografias e livros), informativos (artigos, notícias, relatórios e reportagens) e instrucionais ou normativos (apostilas, perguntas e respostas de fóruns, guias, manuais, decretos, normas técnicas, notas técnicas, portarias, tutoriais e documentos)[123].
Tamanho	Cada campo nocional da CSS deverá contar com, no mínimo, 100 mil palavras[124].
Modalidade	Escrita.
Público-alvo	Pesquisadores, aprendizes e profissionais da Computação.
Estado natural dos textos	Formato eletrônico e sem a necessidade de reconhecimento de seus caracteres[125].

Fonte: o autor.

Após estabelecermos o *design* do CoCLI, definimos os recursos financeiros, tecnológicos, materiais e humanos que seriam despendidos para a execução dos projetos. Como todo o trabalho desta pesquisa foi realizado por nós mesmos e hospedamos o *ToGatherUp*, de forma gratuita, no servidor *web* do ILEEL da UFU,

[120] Assunto do *corpus*.
[121] Período de tempo em que os textos do *corpus* foram publicados.
[122] Selecionamos a língua inglesa para a construção do *corpus*, pois, de acordo com Swales (1990), a maioria dos materiais publicados na Área da Computação está nesse idioma.
[123] Definimos os gêneros e tipos textuais pensando que no fato de a Computação ser uma área acadêmica e profissional. Procuramos incluir os textos que julgamos possuir uma maior probabilidade de encontrarmos contextos definitórios e explicativos (PAVEL; NOLET, 2002).
[124] Não identificamos, na literatura da LC, um número padrão estabelecido para um *corpus* ou para as ramificações de uma Árvore de Domínio. Por essa razão, estabelecemos o número de 100 mil palavras como padrão para a nossa pesquisa, partindo do pressuposto de que esse valor é suficiente para a recuperação de informações em uma pesquisa terminológica
[125] Essa condição dos textos facilita a captura deles.

como parte dos projetos do GPELC, sob o domínio www.togatherup.ileel.ufu.br, não foi necessário investirmos recursos financeiros para a sua realização.

No Quadro 8, apresentamos o cronograma geral da nossa pesquisa, já que nele podemos ver a organização das atividades relacionadas aos dois projetos de criação do CoCLI.

Quadro 8 – Cronograma geral da pesquisa

Período de tempo	**Ações**
2° Semestre de 2017	Revisar a Literatura. Participar das disciplinas Metodologia de Pesquisa em Linguística e Linguística Aplicada (PEL001), Teorias Linguísticas (PEL002) e Tópicos em Estudos Analítico Descritivos 1: Novas Tecnologias em Análise Lexical (PEL009F). Levantar ferramentas de suporte à construção manual de *corpora*.
1° Semestre de 2018	Construir o *ToGatherUp*. Planejar e executar os projetos de construção dos *corpora*. Coletar dados para a realização do experimento. Participar da disciplina Tópicos em Estudos Linguísticos: Teoria da Avaliatividade e Linguística de *Corpus* (PEL213C).
2° Semestre de 2018	Submeter a pesquisa ao exame de Qualificação. Realizar o experimento. Analisar os resultados do experimento. Redigir a Dissertação. Apresentar uma comunicação oral sobre o *ToGatherUp* no V Simpósio Nacional de Letras e Linguística (Sinalel)[126]. Participar como ministrante de aula no curso Fundamentos da Linguística de *Corpus*.
1° Semestre de 2019	Finalizar a redação da Dissertação.

Fonte: o autor.

[126] O Sinalel é um evento promovido pelo Programa de Pós-Graduação em Estudos da Linguagem (PPGEL) da Universidade Federal de Goiás/Regional Catalão (UFG/RC).

Com o cronograma geral, encerramos a fase inicial dos projetos de construção do CoCLI e passamos à fase de execução deles, na qual iniciamos o processo de obtenção dos dados dos *corpora*.

A primeira atividade desse processo foi localizarmos textos que pudessem ser incluídos nos *corpora*. Escolhemos os textos com base nas referências dos currículos dos cursos da área da Computação das instituições citadas na nota de rodapé 118. Para a localização dos dados, efetuamos pesquisas na Internet. Não solicitamos permissão para o uso dos textos devido à dificuldade de obtenção de autorização para a grande quantidade de materiais necessária para nossos *corpora*. Além desse motivo, desconsideramos a obtenção da permissão de uso por não termos a intenção de publicizar o CoCLI[127] ao término da construção dele.

Logo após, iniciamos o processo de preparação dos textos para incluí-los nos *corpora.* Obtivemos os textos dos *corpora* em suas fontes originais nos formatos PDF, DOC e HTML e os convertemos para o formato TXT. Ademais, limpamos e normalizamos os textos para que a possibilidade de erros de tokenização fosse reduzida durante o processamento dos *corpora*.

A fim de que os textos pudessem atingir a condição desejada (formato TXT, limpos e normalizados), criamos uma dinâmica de realização simultânea das atividades de conversão, limpeza e normalização que envolveu o uso do *Acrobat XI*[128], do *Microsoft Word* (MW)[129], do *Sublime Text 3*[130], de *scripts* das linguagens de programação *Visual Basic for Applications* (VBA) e *JavaScript Object Notation* (JSON) e de expressões regulares, conforme explicitamos nos próximos tópicos. No decorrer desses estágios, procuramos limpar os textos de modo a remover os elementos citados no Quadro 9 e normalizá-los por meio dos procedimentos descritos no Quadro 10.

[127] Apesar da falta da autorização de uso, pequenas partes dos textos dos *corpora* podem ser utilizadas, conforme aponta Mark Davies, na seção *Frequently Asked Questions* (FAQ) do projeto corpus.byu.edu (corpus.byu.edu/faq.asp).
[128] Disponível em: https://helpx.adobe.com/acrobat/kb/acrobat-10-11-downloads.html. Acesso em: 27 jun. 2018.
[129] Disponível em: https://products.office.com/pt-BR/word. Acesso em: 27 jun. 2018.
[130] Disponível em: https://www.sublimetext.com/. Acesso em: 27 jun. 2018.

Quadro 9 – Procedimentos de limpeza de *corpus*

Procedimentos
(a) Remoção de cabeçalhos e rodapés de páginas.
(b) Remoção de elementos gráficos (figuras, imagens e gráficos).
(c) Remoção de imagens.
(d) Remoção de notas de rodapé e fim[131].
(e) Remoção de números de página.
(f) Remoção de referências bibliográficas.
(g) Remoção de listas (sumários, tabelas, figuras, abreviações e gráficos).
(h) Remoção de tabelas e quadros.
(h) Remoção de títulos e subtítulos.
(i) Remoção de legendas de tabelas, figuras e quadros.

Fonte: o autor.

Quadro 10 – Procedimentos de normalização textual

Procedimentos
(a) Remoção de hifens no final de linha[132] provenientes da formatação dos textos em seus formatos originais.
(b) Remoção de quebras de linhas/parágrafos/páginas/seções.
(c) Remoção de espaços em branco duplicados.
(d) Remoção de marcas de parágrafos/recuos.
(e) Remoção de linhas em branco.
(f) Padronização de hifens, apóstrofos, traços e aspas.

Fonte: o autor.

[131] Optamos por excluir esses elementos dos textos por julgarmos o restante das informações das produções escritas suficiente para os objetivos da pesquisa.

[132] O hífen de final de linha (*end-of-line hyphen* ou *soft hyphen*) é inserido em uma palavra por um programa de edição de texto para fins de formatação, separando-a em duas partes distintas e em duas linhas, sendo que a primeira parte sempre está localizada no final da primeira linha e é seguida pelo hífen. O processo de inserção do hífen de final de linha é conhecido como hifenização e, geralmente, obedece às regras de separação silábica da língua do texto.

3.3.3.1.1 Estágio 1 – Conversão dos textos para o formato DOC através do Acrobat XI

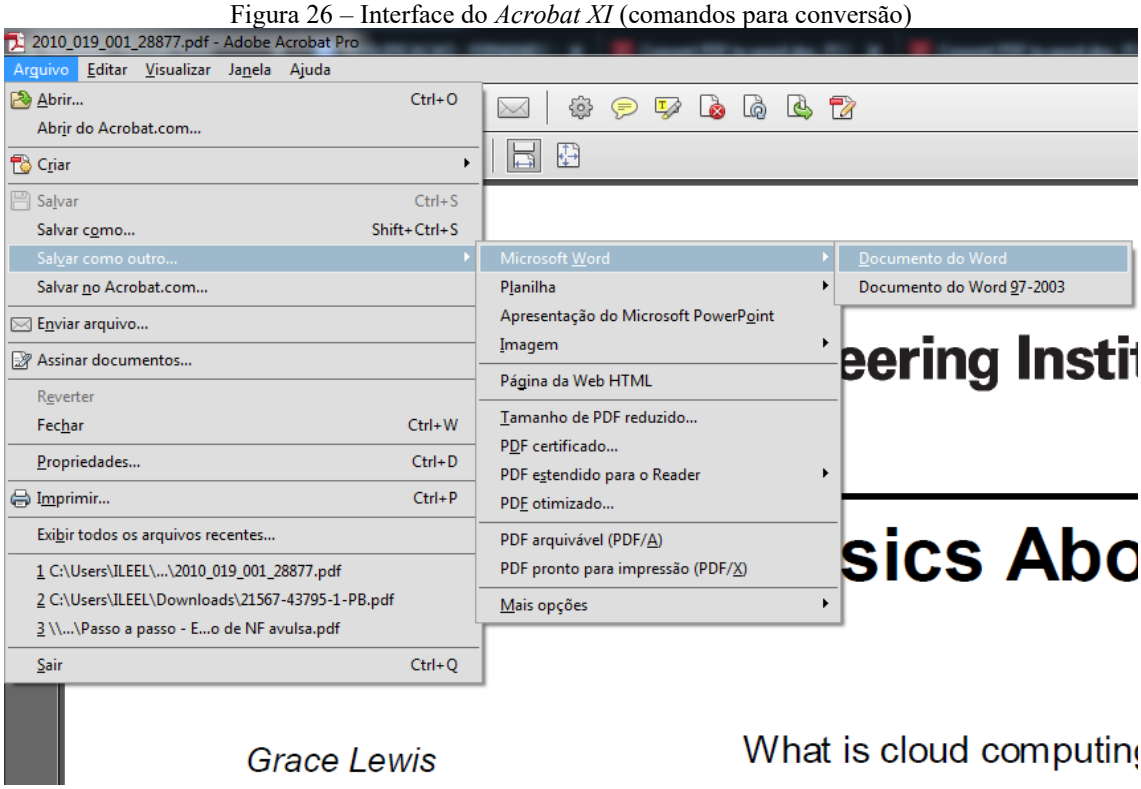
O Estágio 1 consistiu na conversão dos textos em formato PDF para o formato DOC através do *Acrobat XI*, um programa desenvolvido pela *Adobe*[133], pago, mas que pode ser utilizado sem custos por um período de avaliação. Adotamos o *Acrobat XI* por termos acesso à sua licença de uso. Porém, poderíamos ter optado por programas gratuitos, por exemplo, o *Unipdf*[134] ou por serviços de conversão *on-line*, como o *Ilovepdf*[135], para a realização das conversões.

Embora o *Acrobat XI* ofereça a possibilidade de converter arquivos diretamente para o formato TXT, fizemos a conversão intermediária para o formato DOC para que fôssemos capazes de proceder com a limpeza do texto por meio de funções do MW e de *scripts* em VBA no Estágio 2 – referente ao tratamento dos textos e descrito mais adiante. Realizamos a conversão dos textos em formato PDF para o formato DOC, no *Acrobat XI*, seguindo os passos:

1. Após a abertura do texto no *Acrobat XI*, selecionamos a opção "Arquivo" do menu. Em seguida, clicamos em "Salvar como outro...", "*Microsoft Word*" e "Documento do *Word*";
2. Na janela "Salvar como", selecionamos o local de destino do arquivo e, no campo "Nome", mantivemos o nome sugerido pelo programa;
3. Por fim, clicamos no botão "Salvar".

[133] A Adobe (www.adobe.com/br) foi fundada em dezembro de 1982 por Charles Geschke e John Warnock e é mundialmente reconhecida no mercado de programas de edição de imagens, vídeos e textos.
[134] Disponível em: http://unipdf.com. Acesso em: 11 jul. 2018.
[135] Disponível em: www.ilovepdf.com. Acesso em: 11 jul. 2018.

Figura 26 – Interface do *Acrobat XI* (comandos para conversão)



Fonte: *Acrobat XI*.

3.3.3.1.2 Estágio 2 – Limpeza dos textos com o uso de funcionalidades do *Microsoft Word* e de *scripts* em VBA

O Estágio 2 consistiu na realização de parte da limpeza dos textos, em formato DOC, por meio de funções do programa MW[136], desenvolvido pela *Microsoft*, e do uso de *scripts* em VBA. Nessa etapa, tratamos os textos provenientes da conversão realizada no Estágio 1 (convertidos do formato PDF para DOC) e os textos cujo formato original era o DOC. É válido ressaltarmos que as formatações presentes nos textos no formato PDF são mantidas pelo *Acrobat XI* durante a conversão dos textos para o formato DOC. A manutenção das formatações possibilitou a realização de procedimentos de limpeza no MW durante o Estágio 2.

O MW é um aplicativo[137] de processamento de textos (*word-processing application*) projetado para a criação e edição de documentos de texto. O formato de

[136] Utilizamos a versão *desktop* do *Word 2016*, obtida gratuitamente no pacote *Office 365 Education* para alunos e professores de instituições acadêmicas, no endereço eletrônico https://products.office.com/pt-br/student/office-in-education.
[137] Aplicativos são tipos específicos de programas que realizam determinadas tarefas como, por exemplo, o processamento de textos.

arquivo padrão do MW é o DOC ou DOCX e as extensões padrão associadas aos documentos criados no programa são .doc ou .docx.

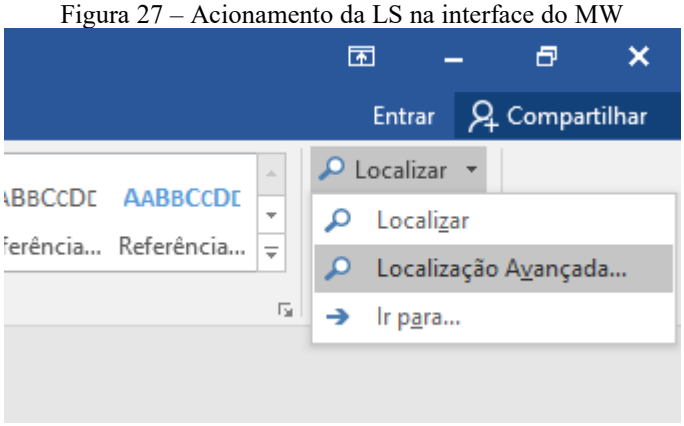
O MW possui um conjunto de funcionalidades que possibilitam a manipulação de textos. Podemos usar a ferramenta para editar, arranjar e aplicar formatações ou, ainda, para configurar os documentos de textos, definindo o tamanho, as margens, o cabeçalho e rodapé de uma página. Além da manipulação textual, o MW oferece funcionalidades para a localização de partes do texto.

O comando Localizar e Substituir (doravante LS) do MW nos permite localizar partes de um texto e substituí-las por outros conteúdos. Podemos classificar as formas de Localizar em dois tipos: com base em conteúdos e com base em formatações. Ao primeiro tipo, atribuímos o nome Localização por Conteúdo (LpC) e, ao segundo, o nome Localização por Formato (LpF).

O LS pode ser acionado, na interface do MW, ao clicarmos na opção "Localização Avançada" da seção "Edição", presente na guia "Página Inicial", conforme a Figura 27, ou por meio do atalho de teclado Ctrl+U[138].

[138] Atalhos de teclado (*keyboard shortcuts*) permitem o acionamento de funcionalidades de programas por meio de combinações de teclas. O atalho Ctrl+U refere-se ao acionamento da tecla Ctrl em conjunto com a tecla U.

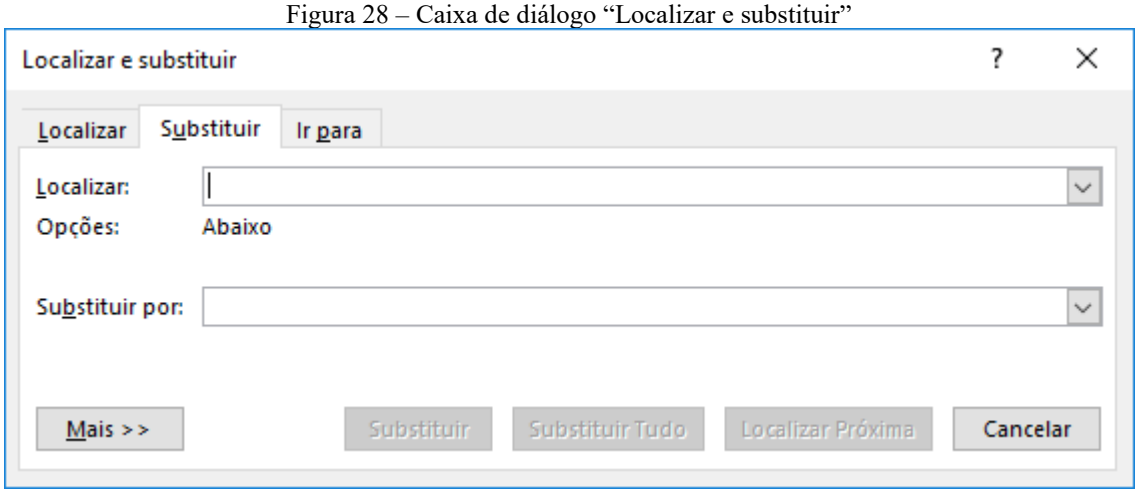
Figura 27 – Acionamento da LS na interface do MW



Fonte: o autor.

Ao acionarmos o comando LS, o MW exibe a caixa de diálogo[139] "Localizar e substituir", ilustrada na Figura 28, com a guia "Substituir" ativada. Para realizarmos a substituição de um conteúdo textual por outro, devemos digitar o conteúdo a ser substituído na caixa de texto[140] "Localizar:", o conteúdo que irá substituí-lo na caixa de texto "Substituir por:" e, em seguida, pressionarmos o botão "Substituir" caso queiramos que a substituição seja feita somente no primeiro elemento correspondente ao conteúdo da caixa de texto "Localizar:", ou o botão[141] "Substituir Tudo" caso desejemos que todas as ocorrências correspondentes ao conteúdo da caixa de texto "Localizar:" sejam substituídas.

Figura 28 – Caixa de diálogo "Localizar e substituir"



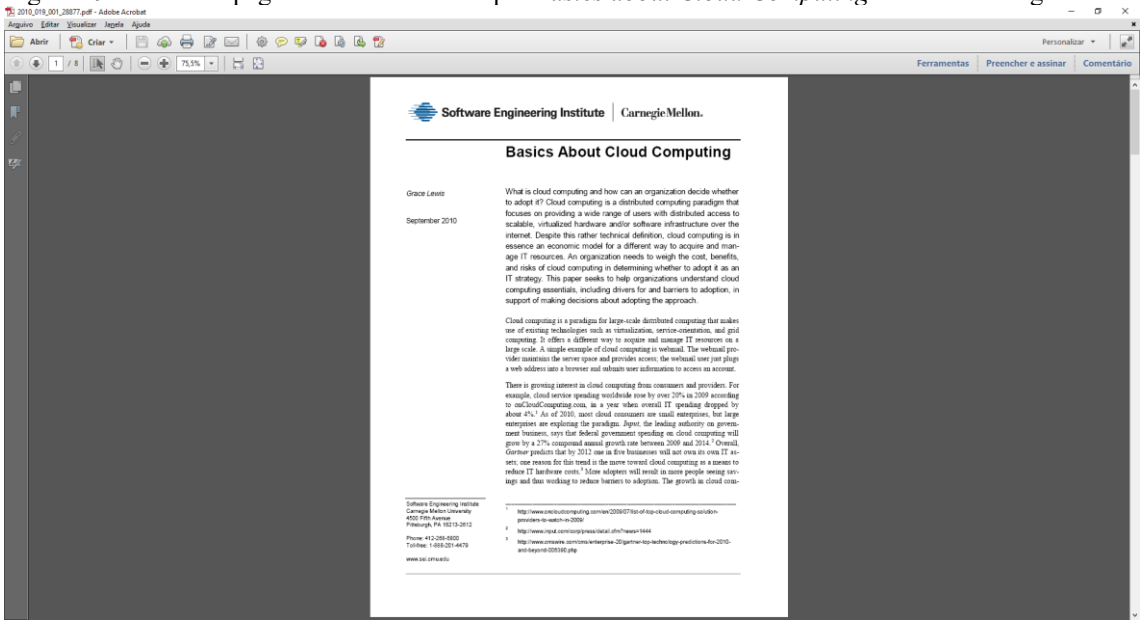
Fonte: o autor.

[139] Caixas de diálogo são elementos de interface dos programas que apresentam controles para a interação com o usuário.
[140] Caixas de texto são elementos de interface dos programas nas quais o usuário pode inserir textos para processamento.
[141] Botões são elementos de interface dos programas que podem ser acionados para a chamada de uma ação (*call to action*).

Ao acionarmos o botão “Mais >>” da caixa de diálogo “Localizar e substituir”, são exibidas opções adicionais para a localização de dados. Dentre as adicionais, ao clicarmos sobre o botão “Formatar”, aparecem as opções referentes à LpF. A realização da limpeza dos textos dos nossos *corpora* com o uso da LS ocorreu, principalmente, com o uso da LpF, que possibilitou a substituição dos elementos textuais indesejados com base nas suas formatações características que os distinguiam do restante do texto. Para a realização das exclusões, criamos um procedimento para a identificação dos padrões de formatação característicos dos elementos textuais indesejados e configuramos a LpF da forma adequada à execução da tarefa.

Para demonstrarmos o método de identificação de padrões de formatação utilizado na nossa pesquisa, descrevemos, por exemplo, o procedimento realizado no texto *Basics about Cloud Computing*, que aparece na Figura 29 em seu estado original (no formato PDF). Na demonstração, contemplamos apenas os elementos da primeira página do texto.

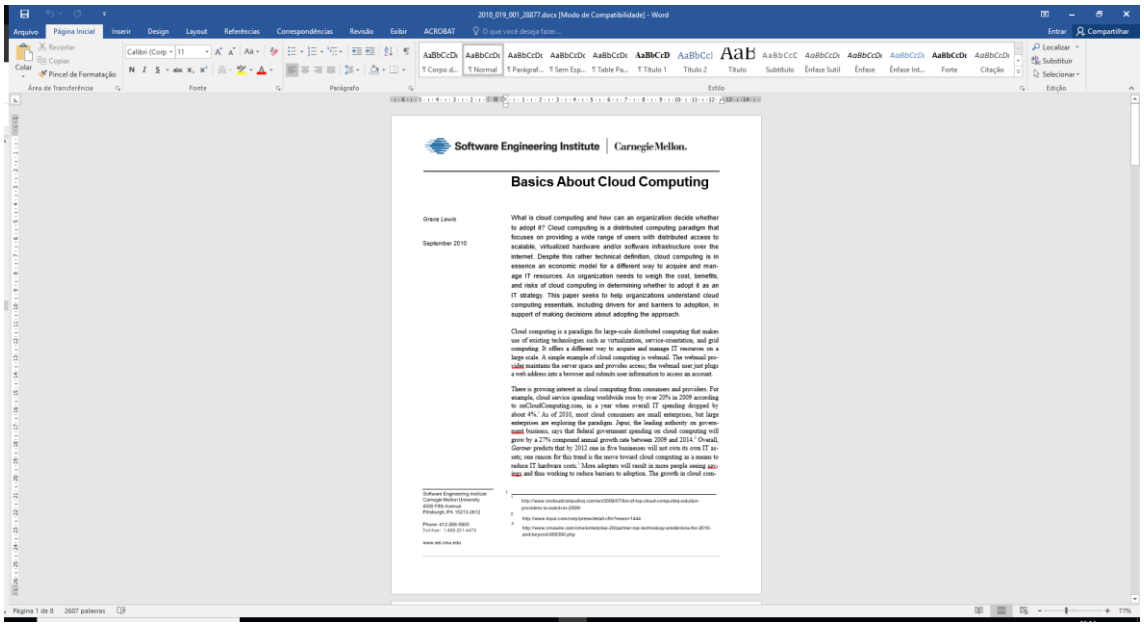
Figura 29 – Primeira página do texto de exemplo *Basics about Cloud Computing* no formato original PDF



Fonte: o autor.

Por questões didáticas, optamos por explicar o procedimento de remoção de elemento com LpF em duas etapas. A primeira envolve a identificação do padrão de formatação do elemento a ser removido. Considerando que o texto de exemplo apresentava-se no formato PDF, inicialmente realizamos a sua conversão para formato DOC, depois, abrimos o texto no MW.

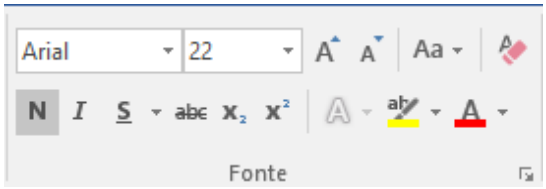
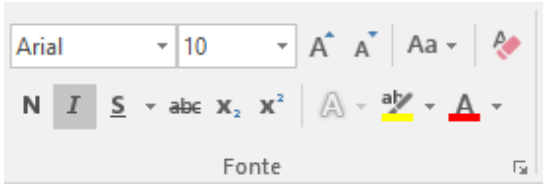
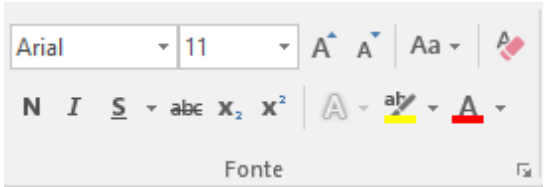
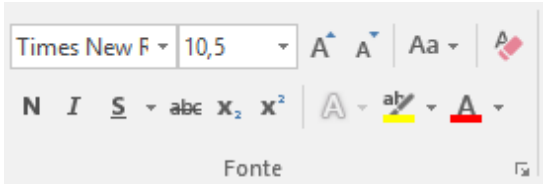
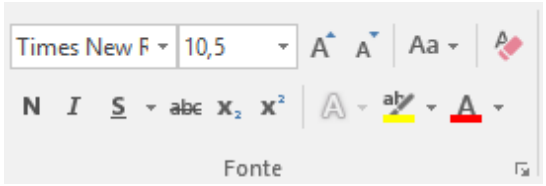
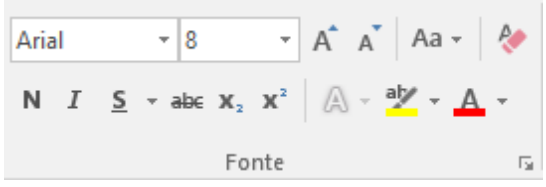
Figura 30 – Primeira página do texto de exemplo *Basics about Cloud Computing* após conversão para o formato DOC

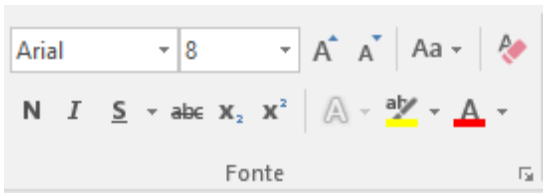


Fonte: o autor.

A Figura 30 nos mostra que a formatação original do texto não foi perdida após a sua conversão. Por isso, ao selecionarmos os elementos, fomos capazes de verificar, na guia “Página Inicial” do MW, quais as formatações que eles apresentavam. O Quadro 11 exibe, na coluna “Formatações”, os recortes da seção “Fonte”, da guia “Página Inicial” do MW, com as formatações de cada um dos elementos identificados na coluna “Elemento”.

Quadro 11 – Elementos e descrição das formatações do texto *Basics about Cloud Computing*

Elemento	Formatações
Elemento 1 – Cabeçalho: 	Por ser uma imagem, não aplicamos a verificação de formatações.
Elemento 2 – Título do texto: **Basics About Cloud Computing**	
Elemento 3 – Autoria e data: *Grace Lewis* September 2010	
Elemento 4 – Primeiro parágrafo: What is cloud computing and how can an organization decide whether to adopt it? Cloud computing is a distributed computing paradigm that focuses on providing a wide range of users with distributed access to scalable, virtualized hardware and/or software infrastructure over the internet. Despite this rather technical definition, cloud computing is in essence an economic model for a different way to acquire and manage IT resources. An organization needs to weigh the cost, benefits, and risks of cloud computing in determining whether to adopt it as an IT strategy. This paper seeks to help organizations understand cloud computing essentials, including drivers for and barriers to adoption, in support of making decisions about adopting the approach.	
Elemento 5 – Segundo parágrafo: Cloud computing is a paradigm for large-scale distributed computing that makes use of existing technologies such as virtualization, service-orientation, and grid computing. It offers a different way to acquire and manage IT resources on a large scale. A simple example of cloud computing is webmail. The webmail provider maintains the server space and provides access; the webmail user just plugs a web address into a browser and submits user information to access an account.	
Elemento 6 – Terceiro parágrafo: There is growing interest in cloud computing from consumers and providers. For example, cloud service spending worldwide rose by over 20% in 2009 according to onCloudComputing.com, in a year when overall IT spending dropped by about 4%.[1] As of 2010, most cloud consumers are small enterprises, but large enterprises are exploring the paradigm. *Input*, the leading authority on government business, says that federal government spending on cloud computing will grow by a 27% compound annual growth rate between 2009 and 2014.[2] Overall, *Gartner* predicts that by 2012 one in five businesses will not own its own IT assets; one reason for this trend is the move toward cloud computing as a means to reduce IT hardware costs.[3] More adopters will result in more people seeing savings and thus working to reduce barriers to adoption. The growth in cloud com-	
Elemento 7 – Notas de rodapé: 1 http://www.oncloudcomputing.com/en/2009/07/list-of-top-cloud-computing-solution-providers-to-watch-in-2009/ 2 http://www.input.com/corp/press/detail.cfm?news=1444 3 http://www.cmswire.com/cms/enterprise-20/gartner-top-technology-predictions-for-2010-and-beyond-006390.php	

Elemento 8 – Informações editoriais:	
Software Engineering Institute Carnegie Mellon University 4500 Fifth Avenue Pittsburgh, PA 15213-2612 Phone: 412-268-5800 Toll-free: 1-888-201-4479 www.sei.cmu.edu	

Fonte: o autor.

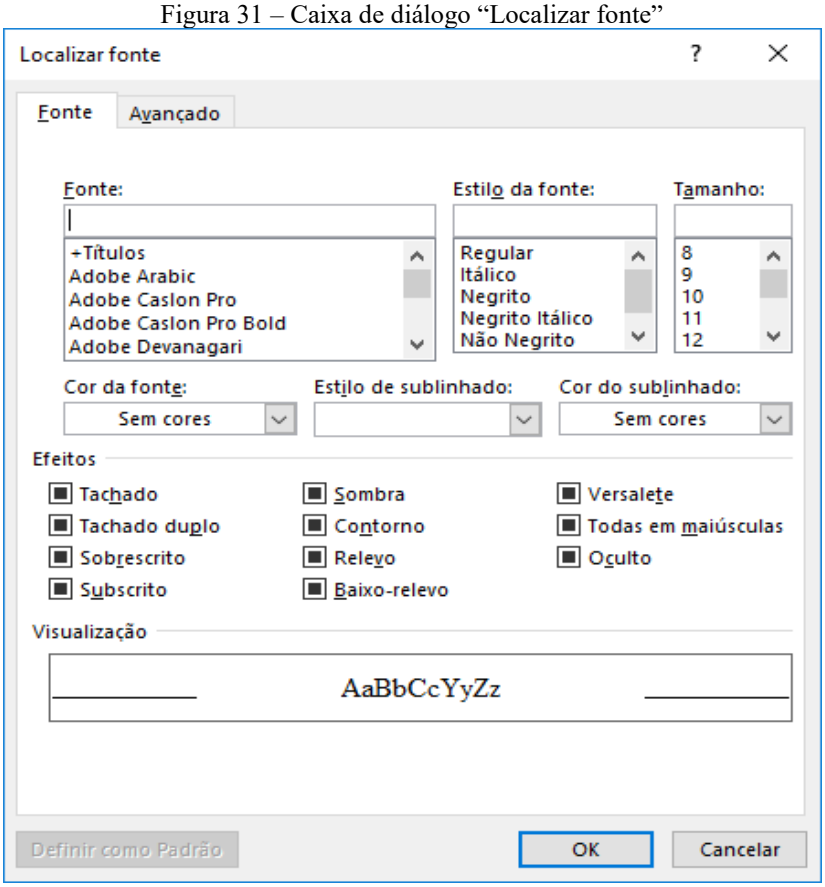
Ao analisarmos os elementos e fontes do Quadro 11, chegamos à conclusão de que:

a) o elemento 2 foi formatado com a fonte *Arial*, tamanho 22 e negrito;
b) o elemento 3 foi formatado com a fonte *Arial*, tamanho 10 e itálico;
c) o elemento 4 foi formatado com a fonte *Arial* e tamanho 11;
d) os elementos 5 e 6 foram formatados com a fonte *Times New Roman* e tamanho 10, 5;
e) os elementos 7 e 8 foram formatados com a fonte *Arial* e tamanho 8.

A segunda etapa alude à retirada de elementos indesejados por meio da LS e da LpF. A identificação dos padrões de formatação dos elementos do texto de exemplo nos permitiu usar o comando LS com base na LpF para a remoção dos elementos que não queríamos no texto. Para demonstrarmos esse procedimento de limpeza, tomamos como exemplo a eliminação dos elementos 7 (Notas de rodapé) e 8 (Informações editoriais*)* do texto *Basics about Cloud Computing*. A seguir, descrevemos os caminhos que percorremos:

1) Acionamos a caixa de diálogo “Localizar e substituir” por meio do comando Ctrl+U e, em seguida, clicamos na opção “Mais >>” para termos acesso à opção “Formatar”;
2) Acionamos a opção “Formatar” e, dentre as opções exibidas, selecionamos “Fonte…”. O MW exibiu a caixa de diálogo “Localizar fonte” de acordo com a Figura 31;

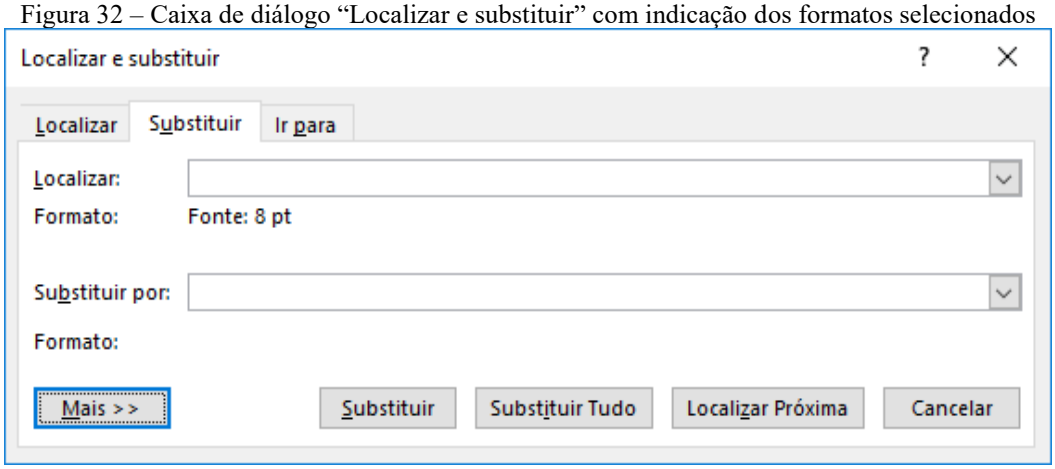
Figura 31 – Caixa de diálogo “Localizar fonte”



Fonte: MW.

3) Na guia “Fonte”, selecionamos as formatações correspondentes ao padrão dos elementos que identificamos anteriormente, ou seja, no campo “Fonte”, colocamos *Arial* e, no campo “Tamanho”, escolhemos 8. Em seguida, clicamos no botão “*OK*”;
4) O MW exibiu a caixa de diálogo “Localizar e substituir”, com a indicação, no campo “Localizar”, dos formatos selecionados no passo anterior, conforme a Figura 32:

Figura 32 – Caixa de diálogo “Localizar e substituir” com indicação dos formatos selecionados



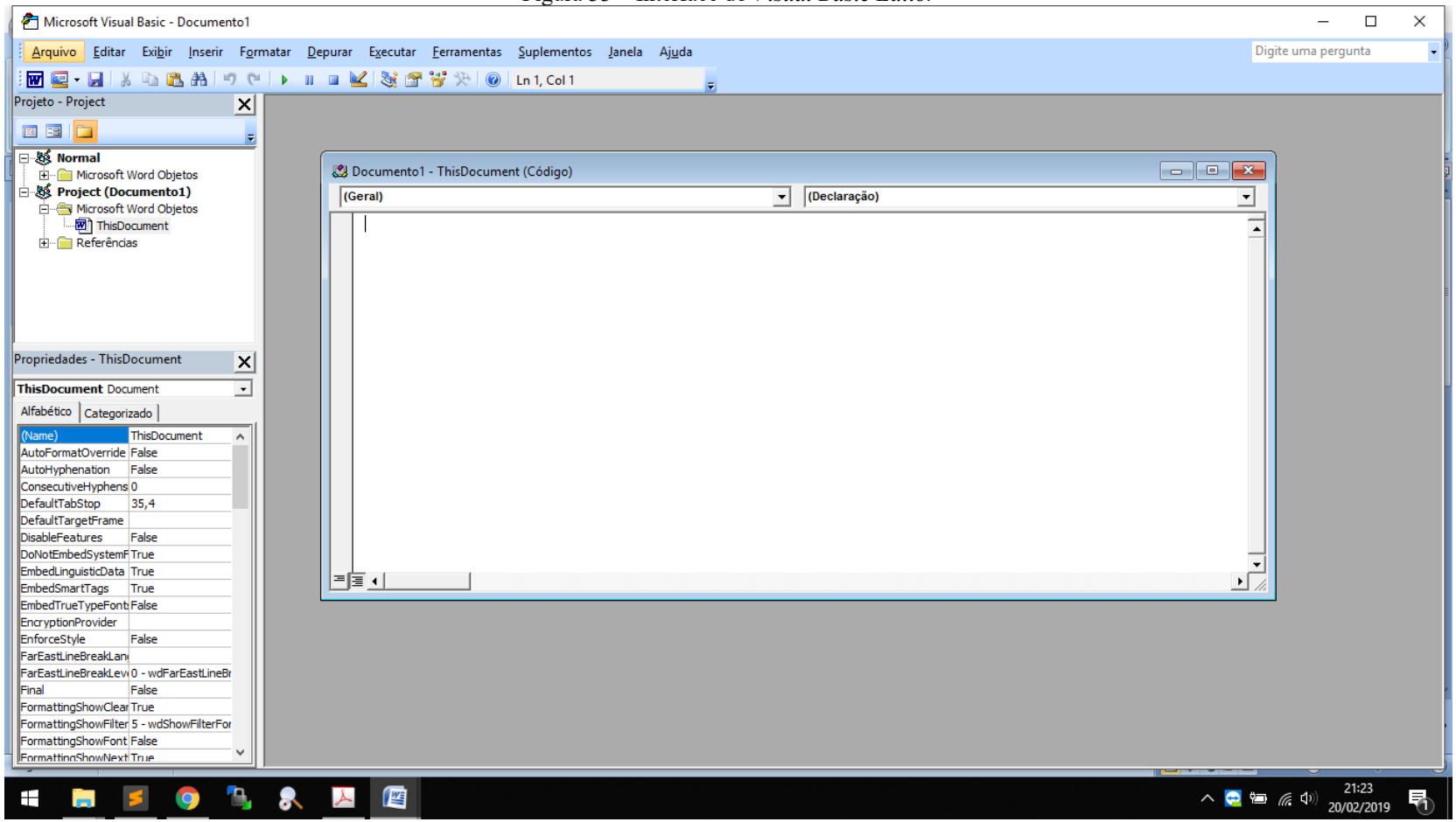
Fonte: MW.

5) Deixamos o campo “Substituir por” vazio e clicamos no botão “Substituir Tudo”.

Ao efetuarmos o procedimento 5, solicitamos ao MW a substituição de todos os conteúdos do texto que possuíam a fonte *Arial* e o tamanho 8 por nada. Em outras palavras, fizemos com que o aplicativo removesse todas as notas de rodapé e informações editoriais do texto. O método de remoção de elementos apresentado pode ser aplicado a outros elementos do texto desde que eles apresentem formatações características que permitam contrastá-los dos demais elementos do texto. A utilização do método citado é eficaz, mas exige do pesquisador o cuidado de certificar-se de que elementos que devam ser mantidos no texto não possuam as mesmas características de formatação de elementos indesejáveis.

O MW faz também a localização e substituição de conteúdos por meio de *scripts* em VBA, que é a linguagem de programação utilizada no MW para a criação de macros que podem ser editadas no *Visual Basic Editor*, uma ferramenta de edição de *scripts* em VBA incorporada ao MW. A Figura 33 exibe a interface do *Visual Basic Editor*, que pode ser aberta pelo usuário do MW pelo acionamento da sequência de teclas Alt+F11 no *Windows*.

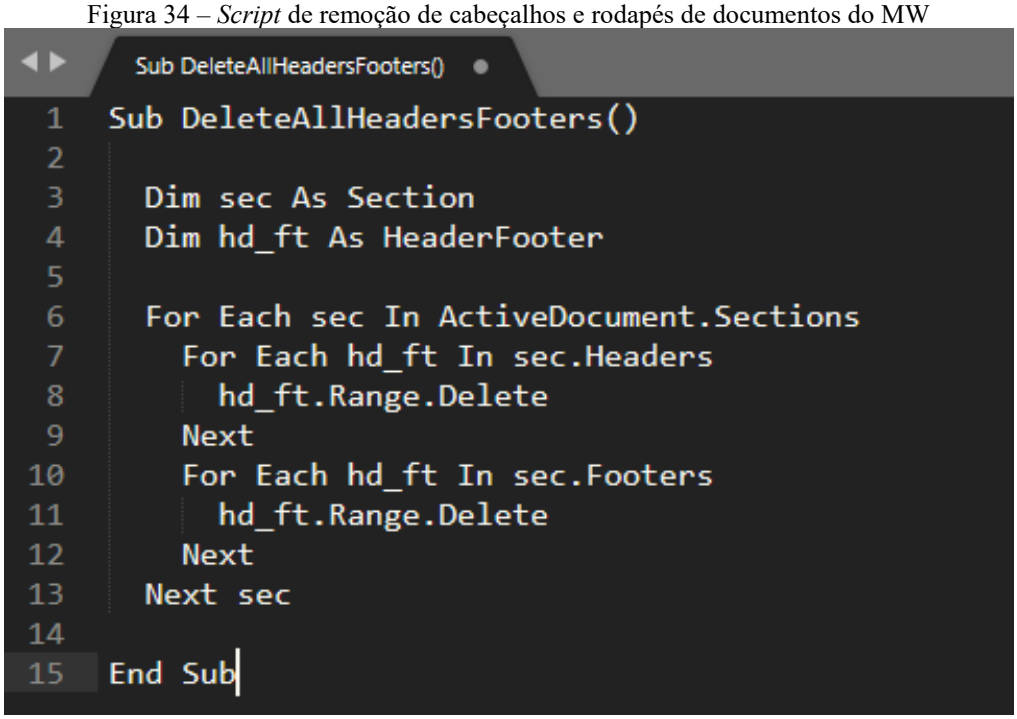
Figura 33 – Interface do *Visual Basic Editor*



Fonte: *Visual Basic Editor*.

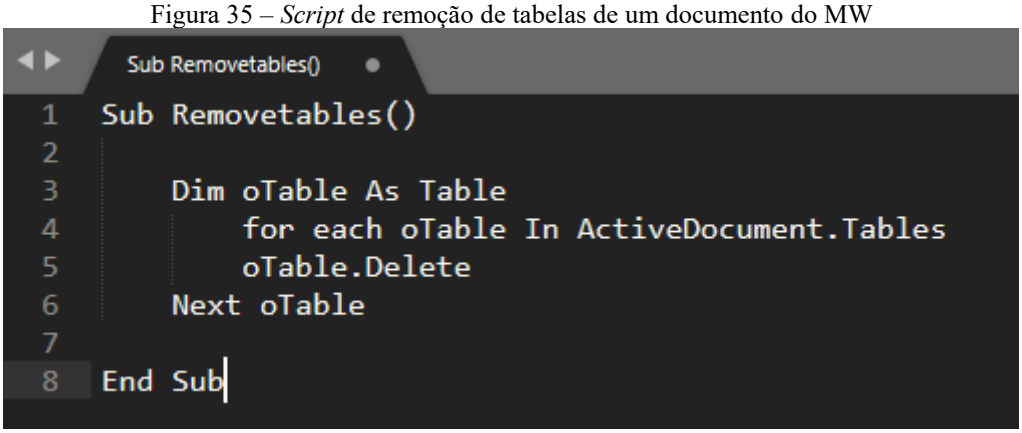
As Figuras 34 e 35 apresentam exemplos[142] de *scripts* em VBA que utilizamos para a limpeza dos textos do CoCLI. A Figura 34 apresenta um *script* simples para a remoção de cabeçalhos e rodapés de documentos do MW e a Figura 35 exibe um *script* para a retirada de todas as tabelas de um documento do MW.

Figura 34 – *Script* de remoção de cabeçalhos e rodapés de documentos do MW



Fonte: *Beyond VBA Tutorial*[143].

Figura 35 – *Script* de remoção de tabelas de um documento do MW



Fonte: o autor.

[142] Os exemplos apenas demonstram o uso do VBA e não trazem explicações sobre a linguagem de programação ou esclarecimentos sobre os seus códigos. A compreensão do VBA exige um estudo aprofundado da linguagem e foge do escopo da nossa pesquisa.

[143] Disponível em: http://vba.relief.jp/word-macro-delete-all-headers-and-footers-active-document/. Acesso em: 20 fev. 2019.

3.3.3.1.3 Estágio 3 – Conversão dos textos para o formato TXT e realização de limpeza e normalização deles no *Sublime Text 3*

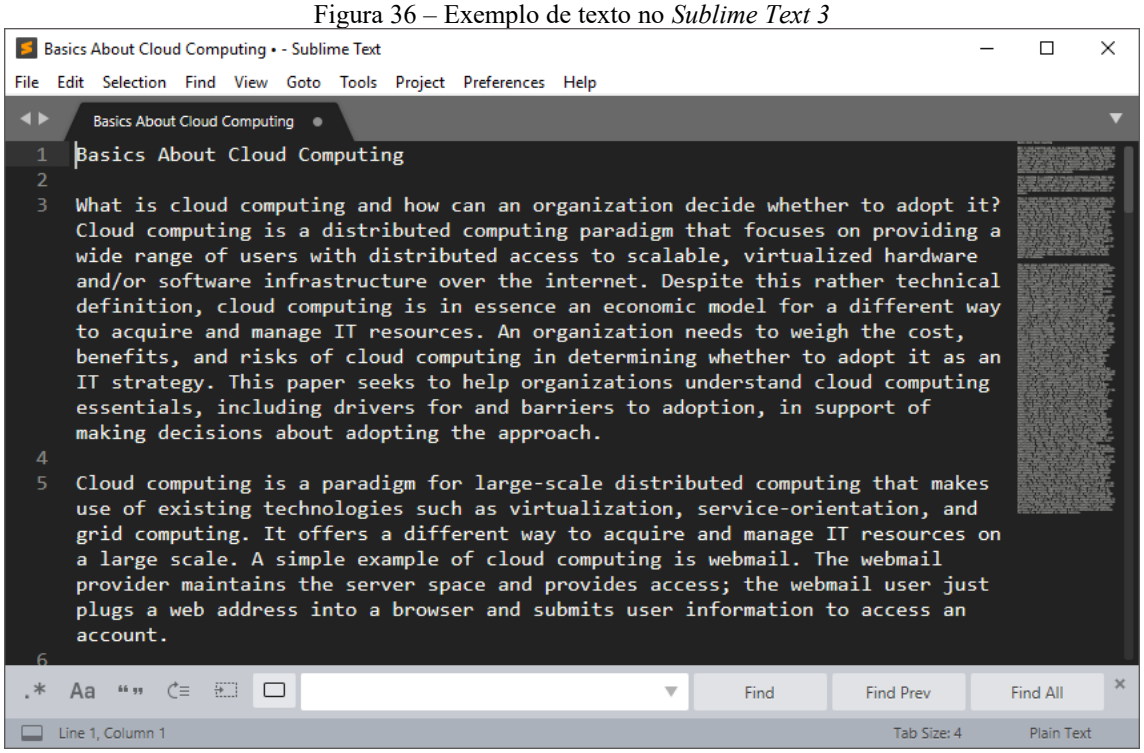
No terceiro estágio da nossa dinâmica de conversão e tratamento de textos, os textos do CoCLI obtidos em PDF e convertidos para o formato DOC (no estágio 2), finalmente, foram convertidos para o formato TXT.

A conversão intermediária (para o formato DOC) foi necessária para que pudéssemos aplicar os tratamentos do MW descritos no estágio anterior. O uso dessa estratégia de conversão, limpeza e normalização fez com que os textos em PDF chegassem ao formato TXT praticamente limpos. Salientamos que os textos obtidos no formato DOC já se apresentavam na condição adequada para os tratamentos do MW e que os textos em HTML foram convertidos diretamente para o formato TXT[144].

Fizemos a conversão dos textos para o formato TXT por meio de comandos de seleção, cópia[145] e colagem[146] dos conteúdos dos arquivos no formato DOC ou HTML para a interface do *Sublime Text 3*, seguidos do comando de salvamento[147] nesse programa. A Figura 36 mostra o aspecto de um texto após a sua colagem no *Sublime Text 3*.

[144] A remoção de elementos indesejáveis dos textos em HTML é mais prática quando feita sem o intermédio do MW devido. A praticidade decorre da eliminação automática das formatações durante a transposição de um texto HTML para o *Sublime Text 3* e da ausência de elementos como os hifens de final de linha e dos números de página nesse formato.
[145] Selecionamos todo o conteúdo do arquivo e, em seguida, o copiamos.
[146] Colamos o conteúdo do arquivo no *Sublime Text 3*.
[147] Durante o salvamento dos dados no *Sublime Text 3*, selecionamos o formato TXT para o arquivo.

Figura 36 – Exemplo de texto no *Sublime Text 3*



Fonte: o autor.

O *Sublime Text 3* é um sofisticado editor de texto, popular entre os programadores (OTÁVIO, 2018) e que, na nossa opinião, é superior aos editores de texto simples como o Bloco de Notas (*Notepad*) do *Windows*. A superioridade do *Sublime Text 3* em relação aos editores de textos simples reside na possibilidade de instalação de *plugins* que expandem as funcionalidades da ferramenta, dando margem, por exemplo, à execução de cadeias pré-definidas de substituições de textos e ao uso de expressões regulares para a localização de dados a serem substituídos ou eliminados nos textos.

Assim, lançamos mão dos recursos oferecidos pelo *Sublime Text 3* para a realização da limpeza e normalização dos textos do CoCLI através do uso de um *plugin*, que permitiu a execução de cadeias pré-definidas de localização e substituição de textos e o uso de expressões regulares criadas para a seleção e eliminação de partes indesejadas dos textos.

Ao longo da atividade de limpeza e normalização de textos, percebemos a recorrência de problemas, como a existência de espaços duplicados e sinais escritos de maneiras diferentes (hifens, apóstrofos, traços e aspas). Com o objetivo de evitar a repetição de comandos de localização e substituição para cada um desses problemas, utilizamos o *plugin RegReplace*[148], que possibilitou a criação de um *script* (cf. quadro 12), no formato JSON, para a localização e substituição simultânea de sinais e espaços em branco, isto é, a execução

[148] Disponível em: https://packagecontrol.io/packages/RegReplace. Acesso em: 20 fev. 2019.

de cadeias pré-definidas de localização e substituição de texto promoveu a normalização dos textos[149].

Quadro 12 – *Script* do *RegReplace*

Parte 1[150]

```
{
 "replacements": {
 "replace_traco": {
  "find": "—",
  "replace": "-",
  "greedy": true,
  "case": false
 },
 "replace_abre_aspas": {
  "find": "“",
  "replace": "\"",
  "greedy": true,
  "case": false
 },
 "replace_fecha_aspas": {
  "find": "”",
  "replace": "\"",
  "greedy": true,
  "case": false
 },
 "replace_apostrofo_close": {
  "find": "’",
  "replace": "'",
  "greedy": true,
  "case": false
 },
 "replace_apostrofo_open": {
  "find": "‘",
  "replace": "'",
  "greedy": true,
  "case": false
 },
 "replace_traco_2": {
  "find": "–",
  "replace": "-",
  "greedy": true,
  "case": false
 },
```

Parte 2[151]

```
 [
  {
  "caption": "RegReplace: Replace non Ascii",
  "command": "reg_replace",
  "args": {
   "replacements": [
    "replace_traco",
    "replace_abre_aspas",
    "replace_fecha_aspas",
    "replace_apostrofo_open",
    "replace_apostrofo_close",
    "replace_traco_2",
    "replace_x"
   ]
  }
  },
  {
  "caption": "RegReplace: Replace duplicated spaces",
  "command": "reg_replace",
  "args": {
   "replacements": [
    "replace_spaces_2",
    "replace_spaces_3",
    "replace_spaces_4",
    "replace_spaces_5",
    "replace_spaces_2"
   ]
  }
  },
  {
  "caption": "RegReplace: Replace breaklines for spaces",
  "command": "reg_replace",
  "args": {
   "replacements": [
    "replace_breaklines"
```

[149] O *script* pode ser expandido para a normalização de outros elementos como siglas e nomes, por exemplo.
[150] Inserida no arquivo *reg_replace_rules.sublime-settings*.
[151] Inserida no arquivo *Default.sublime-commands*.

```
  "replace_x": {
   "find": "×",
   "replace": "x",
   "greedy": true,
   "case": false
  },
  "replace_spaces_2": {
   "find": "  ",
   "replace": " ",
   "greedy": true,
   "case": false
  },
  "replace_spaces_3": {
   "find": "   ",
   "replace": " ",
   "greedy": true,
   "case": false
  },
  "replace_spaces_4": {
   "find": " ",
   "replace": " ",
   "greedy": true,
   "case": false
  },
  "replace_spaces_5": {
   "find": "    ",
   "replace": " ",
   "greedy": true,
   "case": false
  },
  "replace_breaklines": {
   "find": "\n",
   "replace": " ",
   "greedy": true,
   "case": false
  },
  "replace_tab": {
   "find": "\t",
   "replace": " ",
   "greedy": true,
   "case": false
  },

  }
}
```

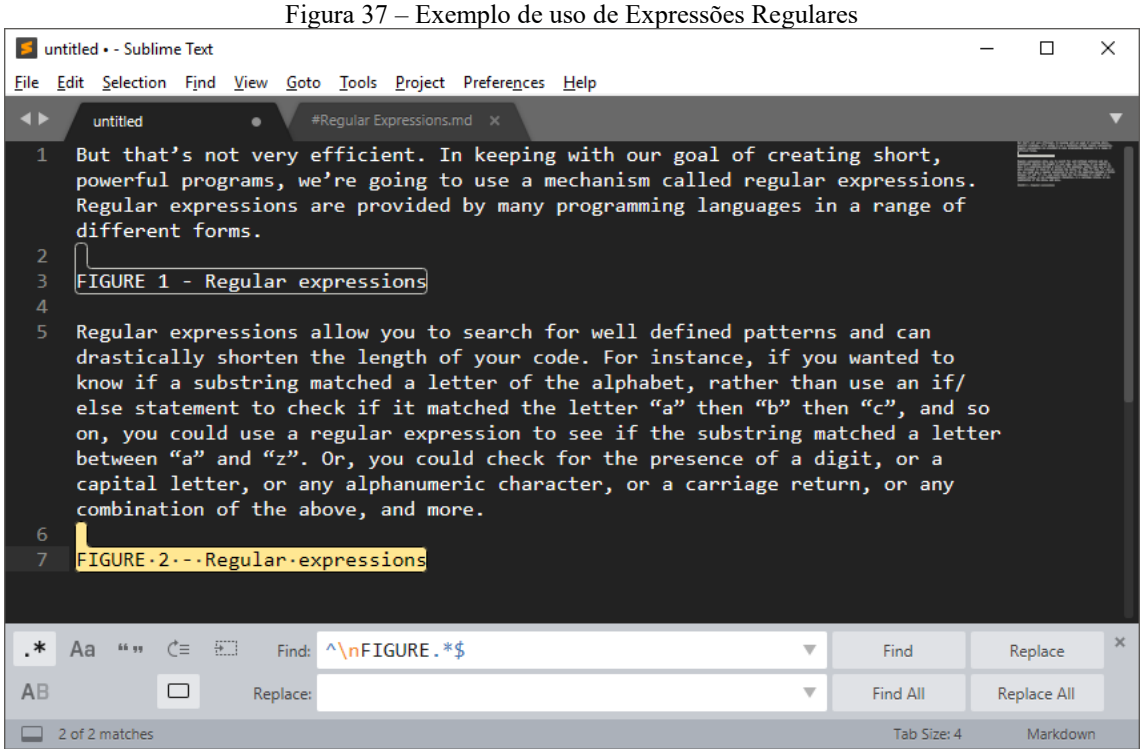
```
    ]
   }
   },
   {
   "caption": "RegReplace: Clean text",
   "command": "reg_replace",
   "args": {
    "replacements": [
     "replace_breaklines",
     "replace_spaces_2",
     "replace_spaces_3",
     "replace_spaces_4",
     "replace_spaces_5",
     "replace_spaces_2",
     "replace_tab",
     "replace_traco",
     "replace_abre_aspas",
     "replace_fecha_aspas",
     "replace_apostrofo_open",
     "replace_apostrofo_close",
     "replace_traco_2",
     "replace_x"
    ]
   }
   },

]
```

Fonte: o autor.

O *Sublime Text 3* possui uma funcionalidade para a localização e substituição de texto semelhante à de outros editores de texto. No entanto, comandos simples de localização e substituição com base em sequência de caracteres podem não ser suficientes para a localização e remoção de partes de textos. A funcionalidade de localização e substituição de texto do *Sublime Text 3* tem a capacidade de processar um mecanismo (TURKEL; CRYMBLE, 2012), chamado Expressões Regulares, que pode resolver as situações em que os simples comandos de localização e substituição não são suficientes para a limpeza de um texto. De acordo com Michael (2012, p. 16), as Expressões Regulares são "*strings*[152] de texto especialmente codificadas e utilizadas como padrões para corresponder a conjuntos de *strings*".

A Figura 37 apresenta um exemplo[153] de Expressão Regular que utilizamos para selecionar todas as linhas de um texto que começavam com a palavra "*FIGURE*".

Figura 37 – Exemplo de uso de Expressões Regulares



Fonte: o autor.

Na Figura 37, observamos a Expressão Regular "^\nFIGURE.*$" no campo "*Find*" da funcionalidade de localização e substituição de textos do *Sublime Text 3*. Ao executarmos o

[152] O termo *string* é utilizado por programadores para designar uma cadeia de caracteres que podem representar uma palavra ou conjuntos de palavras.
[153] Os exemplos apenas demonstram o uso das expressões e não trazem informações sobre a construção das Expressões Regulares. O entendimento das Expressões Regulares depende do estudo aprofundado delas e está fora do escopo da nossa pesquisa.

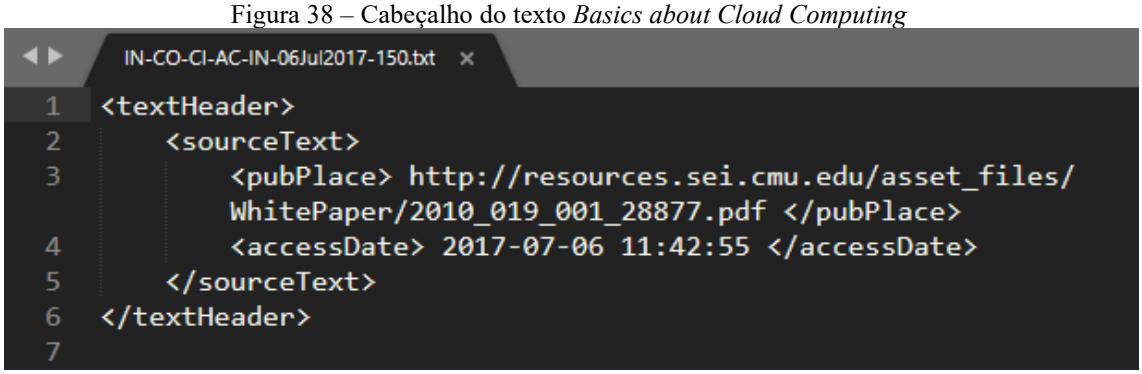
comando de localização e substituição com o uso dessa expressão, o *Sublime Text 3* selecionou todas as linhas iniciadas com a palavra "*FIGURE"* ao longo do texto. Com isso, foi possível eliminarmos as linhas selecionadas (também seria possível substituí-las por outro conteúdo). Ressaltamos que, do mesmo modo que as linhas do nosso exemplo foram removidas, as Expressões Regulares podem ser utilizadas para a eliminação simultânea das partes indesejadas do texto que possuam o padrão descrito em uma Expressão Regular.

Ao concluirmos a conversão, limpeza e normalização dos textos, iniciamos a execução da última atividade do processo de preparação dos *corpora* – o enriquecimento dos dados. Esta e as relativas ao armazenamento compõem o conjunto de atividades que tiveram a forma de execução completamente alterada com a incorporação do *ToGatherUp*. No próximo tópico, descrevemos como cada uma delas foi realizada no Método 1 e 2.

3.3.3.2 Enriquecimento e armazenamento dos dados: diferenças entre os métodos 1 e 2

O enriquecimento dos dados dos *corpora* da nossa pesquisa consistiu na inserção de cabeçalhos construídos a partir de metadados dos textos – fonte (origem do texto) e data de inclusão (data e hora em que o texto foi incluído nos *corpora*). Os cabeçalhos foram construídos de acordo com os princípios da linguagem XML (cf. tópico 2.4.2.2 O processo de preparação dos dados do *corpus* do segundo capítulo) e, no caso do texto *Basics about Cloud Computing*, apresentou a estrutura ilustrada na Figura 38.

Figura 38 – Cabeçalho do texto *Basics about Cloud Computing*



Fonte: o autor.

No Método 1, construímos o cabeçalho da Figura 38 e o inserimos no arquivo do texto de forma manual. Já no Método 2, programamos o *ToGatherUp* para ele fosse capaz de construir e inserir, automaticamente, o cabeçalho no texto. A construção do cabeçalho pelo *ToGatherUp* ocorreu de acordo com a estrutura XML que definimos. O *ToGatherUp* utilizou

os metadados oriundos do seu banco de dados para alimentar a estrutura do cabeçalho, que foi inserido no início de cada texto (a posição de inserção, também, foi realizada em conformidade com o critério que estabelecemos).

A inclusão dos cabeçalhos encerrou o processo de preparação dos textos, que foi sucedido pelas atividades de armazenamento dos dados dos *corpora*. Para a nomeação dos arquivos dos textos, utilizamos a convenção de nomeação de arquivos apresentada no tópico 3.2.2.2 Cadastro de Textos deste capítulo. No Método 1, nomeamos os textos dos *corpora* manualmente, o que nos exigiu muita atenção, esforço e constante consulta às regras de construção da convenção. No Método 2, todo o trabalho foi executado de maneira automática pelo *ToGatherUp* durante o registro do texto no Cadastro de Textos da ferramenta. Para que isso fosse possível, programamos o *ToGatherUp* para nomear os arquivos de acordo com as regras da convenção de nomeação de arquivos que adotamos e com os metadados dos textos registrados em seu banco de dados.

A última atividade dos projetos de construção do CoCLI foi o salvamento (arquivamento) dos textos. No Método 1, salvamos manualmente os arquivos dos *corpora* nos diretórios correspondentes às áreas e subáreas presentes na Árvore de Domínio da Computação. Executamos essa atividade com o *Windows Explorer* do *Windows.* No *Windows Explorer*, criamos uma pasta para cada área e subárea da Árvore de Domínio da Computação, obedecendo às suas relações hierárquicas, e salvamos os arquivos em seus respectivos diretórios. No Método 2, o *ToGatherUp* armazenou automaticamente os arquivos em consonância com os princípios e a funcionalidade de armazenamento da ferramenta apresentados no tópico 3.2.2.2 Cadastro de Textos deste capítulo. No contexto do *ToGatherUp*, o local dos arquivos é indiferente, uma vez que a ferramenta é capaz de organizar os arquivos de acordo com a consulta estabelecida pelo usuário do sistema. No caso do CoCLI, programamos o *ToGatherUp* para exportar os arquivos conforme a estrutura da Árvore de Domínio da Computação.

3.4 O experimento

O objetivo principal desta pesquisa é determinar os efeitos da incorporação do *ToGatherUp* no esforço necessário para a construção manual de *corpora*. A abordagem que utilizamos para alcançar tal propósito foi a realização de um experimento de comparação entre o ETP do CoCLI construído com o Método 1 (sem o *ToGatherUp*) e o ETP do CoCLI elaborado com o Método 2 (com o *ToGatherUp*).

Com o fito de obtermos os dados necessários para colocarmos em prática o experimento, em um primeiro momento, desenvolvemos um conjunto de métricas de medição de esforço – EA, ETP e ETCT – mencionado na seção 3.1 A medida do esforço, deste capítulo. Na sequência, executamos os projetos de construção das duas versões idênticas do CoCLI e, durante essa tarefa, procedemos com a tabulação manual do EA, fornecido pelo cronômetro do *ToGatherUp* (Instrumento 1), de cada uma das atividades necessárias para a inclusão dos textos dos *corpora*. Usamos uma planilha do *Google* (Instrumento 2) para nos auxiliar na organização dessas informações.

Do conjunto de dados (*dataset*) sobre os EAs dos projetos de construção do CoCLI registrados no Instrumento 2, extraímos uma amostra aleatória com as informações referentes a 50 textos. Os dados da amostra foram submetidos a um experimento que, por meio de um teste estatístico, produziu as informações que nos permitiram determinar o efeito da incorporação do *ToGatherUp* na construção manual das duas versões do CoCLI. Nos próximos tópicos, apresentamos como realizamos o experimento.

3.4.1 T-Test

De acordo com Rumsey (2010), testar uma hipótese é uma tentativa de se "confirmar ou negar uma declaração sobre uma população[154] a partir dos dados de sua amostra[155]" (RUMSEY, 2010, p. 87)[156]. Para a autora, quando um teste de hipóteses[157] envolve a comparação entre parâmetros numéricos, o objeto de interesse é a diferença entre as médias[158] (*means*) desses parâmetros. Como a nossa análise envolveu a comparação entre o ETP dos diferentes projetos de construção do CoCLI (dois parâmetros numéricos), utilizamos um teste de hipóteses conhecido como *T-Test* que, segundo Dodge (2008), é apropriado para testar hipóteses a partir da comparação entre as médias de duas populações em que os elementos de uma delas possuem uma relação com os elementos da outra.

Correia (2003) afirma que a realização de um teste de hipóteses envolve os seguintes passos: a) o levantamento de uma amostra da população; b) a definição das hipóteses do teste

[154] Para Correia (2003), população é "uma coleção completa de todos os elementos a serem estudados" (CORREIA, 2003, p. 9).
[155] Consoante Correia (2003), amostra é "uma subcoleção de elementos extraídos de uma população" (CORREIA, 2003, p. 9).
[156] Original: "*trying to confirm or deny a claim about a population using data from a sample*".
[157] Segundo Correia (2003), um teste de hipóteses é "técnica para se fazer inferência estatística. Ou seja, a partir de um teste de hipóteses realizado com os dados amostrais, pode-se fazer inferências sobre a população" (CORREIA, 2003, p. 100).
[158] Conforme Correia (2003), uma média ou média aritmética é "o quociente da divisão da soma dos valores da variável pelo número deles" (CORREIA, 2003, p. 49).

(nula e alternativa[159]) e c) a definição de um nível de significância[160]. Nos próximos tópicos, apresentamos como colocamos em prática a orientação de Correia (2003).

3.4.2 O levantamento da amostra da população

Para a realização do *T-Test*, importamos os dados tabulados no Instrumento 2[161] no *software Statistics Statistical Package for the Social Sciences* (SSPS)[162], uma ferramenta de análise estatísticas, desenvolvida pela IBM, amplamente usada em pesquisas acadêmicas que envolvem a realização de testes estatísticos. Na sequência, utilizamos uma função do SSPS para criar uma amostra aleatória[163] de cinquenta textos do CoCLI apresentada na Tabela 1.

Tabela 1 – Amostra da população

ID	Nome do arquivo	ETCT – Método 1[164]	ETCT – Método 2
1	IN-CO-IF-AT-IN-17Jun2017-1.txt	528	393
5	IN-CO-IF-AT-IN-17Jun2017-5.txt	477	341
24	IN-CO-CI-CL-LI-20Jun2017-24.txt	1113	976
60	IN-CO-CI-CL-LI-28Jun2017-60.txt	1095	968
74	IN-CO-CI-CL-LI-28Jun2017-74.txt	1319	1190
97	IN-CO-CI-CL-LI-29Jun2017-97.txt	1964	1833
115	IN-CO-CI-CL-LI-29Jun2017-115.txt	2749	2612
161	IN-CO-IF-CL-IN-08Jul2017-161.txt	418	291
183	IN-CO-IF-AT-IN-10Jul2017-183.txt	404	277
207	IN-CO-IS-CL-LI-11Jul2017-207.txt	949	814
218	IN-CO-IS-CL-LI-11Jul2017-218.txt	2155	2019

[159] Neste capítulo, na seção 3.4.3 A definição das hipóteses do teste, apresentamos os conceitos de hipótese nula e hipótese alternativa.
[160] Neste capítulo, no tópico 3.4.4 A definição de nível de significância, discorremos sobre o conceito de nível de significância.
[161] Os dados correspondem a um total de 791 textos do CoCLI resultante da aplicação dos métodos 1 e 2 de construção. O conjunto de dados apresenta o ETCT de cada um dos métodos.
[162] Utilizamos o SSPS porque a ferramenta realiza os cálculos estatísticos de forma automática. Disponível em: https://www.ibm.com/br-pt/products/spss-statistics. Acesso em: 23 fev. 2019.
[163] Escolhemos os registros que compuseram o conjunto de dados criado de forma automática e aleatória pelo SSPS.
[164] Os valores do ETCT são expressos em segundos.

223	IN-CO-IF-AT-IN-14Jul2017-223.txt	433	303
238	IN-CO-IF-AT-IN-17Jul2017-238.txt	1621	1487
242	IN-CO-IS-GU-IN-18Jul2017-242.txt	832	697
244	IN-CO-IF-AT-IN-18Jul2017-244.txt	444	312
246	IN-CO-IF-AT-IN-18Jul2017-246.txt	2177	2056
249	IN-CO-IF-AT-IN-18Jul2017-249.txt	1396	1259
292	IN-CO-IF-AT-IN-25Jul2017-292.txt	935	809
298	IN-CO-IF-AT-IN-25Jul2017-298.txt	641	510
313	IN-CO-IF-AT-IN-26Jul2017-313.txt	578	452
344	IN-CO-IF-AT-IN-27Jul2017-344.txt	528	401
389	IN-CO-IS-CL-LI-28Jul2017-389.txt	1117	985
391	IN-CO-IS-CL-LI-28Jul2017-391.txt	1497	1361
408	IN-CO-IS-CL-LI-29Jul2017-408.txt	2199	2069
428	IN-CO-IS-DC-IN-31Jul2017-428.txt	659	524
437	IN-CO-IF-AT-IN-31Jul2017-437.txt	572	444
440	IN-CO-IF-AT-IN-31Jul2017-440.txt	1151	1023
469	IN-CO-IS-CL-LI-31Jul2017-469.txt	724	591
479	IN-CO-IS-CL-LI-01Aug2017-479.txt	1885	1745
557	IN-CO-IS-CL-LI-04Aug2017-557.txt	2106	1973
560	IN-CO-IF-AT-IN-06Aug2017-560.txt	761	626
595	IN-CO-IF-AT-IN-09Aug2017-595.txt	414	286
604	IN-CO-IS-CL-LI-11Aug2017-604.txt	2531	2398
610	IN-CO-IS-CL-LI-11Aug2017-610.txt	1338	1212
613	IN-CO-IS-CL-LI-11Aug2017-613.txt	1545	1416
625	IN-CO-IS-CL-LI-14Aug2017-625.txt	2064	1942
641	IN-CO-IS-CL-LI-15Aug2017-641.txt	2233	2104
679	IN-CO-IS-AT-LI-15Aug2017-679.txt	1143	1010
683	IN-CO-IS-CL-LI-15Aug2017-683.txt	347	217

689	IN-CO-IS-CL-LI-16Aug2017-689.txt	1937	1807
695	IN-CO-IS-CL-LI-16Aug2017-695.txt	934	799
713	IN-CO-IS-DC-IN-21Aug2017-713.txt	12541	12416
714	IN-CO-IS-LV-LI-22Aug2017-714.txt	18860	18727
726	IN-CO-IF-AT-IN-01Sep2017-726.txt	831	706
744	IN-CO-IF-AT-IN-01Sep2017-744.txt	2851	2721
768	IN-CO-IF-AT-IN-05Sep2017-768.txt	769	641
780	IN-CO-IF-AT-IN-05Sep2017-780.txt	498	358
781	IN-CO-IF-AT-IN-05Sep2017-781.txt	527	394
783	IN-CO-IF-AT-IN-05Sep2017-783.txt	493	362
787	IN-CO-IF-AT-IN-05Sep2017-787.txt	441	310

Fonte: SSPS.

A coluna "ETCT – Método 1" contém o ETCT resultante da aplicação do Método 1 (que abreviamos como ETCT – Método 1) e a coluna "ETCT – Método 2" apresenta o ETCT resultante da aplicação do Método 2 (que passamos a chamar de ETCT – Método 2). Os dados referentes ao ETCT – Método 1 constituem o Grupo de Controle[165] (*control group*) da nossa pesquisa e os dados relativos ao ETCT – Método 2 formam o Grupo Experimental (*treatment group*). O tratamento que diferenciou o Grupo de Controle do Grupo Experimental foi a manipulação dos EAs automatizados pelo *ToGatherUp* no Método 2.

3.4.3 A definição das hipóteses do teste

A nossa hipótese de pesquisa parte da ideia de que a incorporação do *ToGatherUp* em projetos de construção manual de *corpora* reduz o tempo e o esforço despendidos pelo pesquisador para a elaboração deles. Para expressarmos essa hipótese na linguagem estatística, usamos os conceitos de hipótese nula[166] (*null hypothesis*) e hipótese alternativa (*alternate hypothesis*).

[165] De acordo com Rumsey (2010), as amostras que são expostas a condições normais (não recebem tratamento ou recebem um tratamento falso, também chamado de placebo) denominam-se Grupo de Controle. Já as amostras sujeitas a tratamento que afeta seus atributos são chamadas de Grupo Experimental.
[166] Na estatística, a hipótese nula é representada por H_0 e a hipótese alternativa, por H_1.

Segundo Charles Brase e Corrine Brase (2011, p. 411), a hipótese nula ou "hipótese estatística"[167] é a declaração que está sob teste e, geralmente, associa-se a resultados como "não houve efeito", "não houve diferença" ou "nada foi alterado" entre a média calculada para o Grupo de Controle e a média calculada para o Grupo Experimental. A hipótese alternativa[168] é definida pelos autores como qualquer declaração diferente da hipótese nula. De acordo com os conceitos de hipótese nula e alternativa, podemos representar a nossa hipótese de pesquisa, na linguagem estatística, conforme ilustra a Figura 39.

Figura 39 – Hipótese da pesquisa expressa na linguagem estatística

Hipótese nula (H_0): ETCT - Método 2 = ou > ETCT - Método 1

Hipótese alternativa (H_1): ETCT - Método 2 < ETCT - Método 1

Fonte: o autor.

A interpretação da Figura 39 pode ser feita da seguinte maneira: nossa hipótese de pesquisa deve ser rejeitada caso o resultado do *T-Test* revele que o ETCT do método que utiliza o *ToGatherUp* é igual ou maior do que o ETCT do método que não utiliza a ferramenta. Se a hipótese nula for rejeitada, ou seja, se o *T-Test* mostrar que o ETCT do método que utiliza o *ToGatherUp* é menor do que o ETCT do método que não utiliza a ferramenta, a hipótese alternativa deve ser aceita e a nossa hipótese de pesquisa confirmada.

3.4.4 A definição de nível de significância

O resultado de um teste de hipótese é estatisticamente significativo quando a probabilidade de que ele tenha ocorrido por acaso seja muito improvável. Para Rumsey (2010), o nível de significância de um teste de hipótese, também conhecido como *alpha level* (α), é dado pelo *p-value* (*probability value*) que, geralmente, é definido em 0.05[169] ou 0.01.

[167] Para Correia (2003), a hipótese estatística "trata-se de [*i.e.* trata de] uma suposição quanto ao valor de um parâmetro populacional, ou quanto à natureza da distribuição de probabilidade de uma variável populacional" (CORREIA, 2003, p. 100).

[168] Autores como Rumsey (2010) também usam a expressão "hipótese de pesquisa" para referenciar a hipótese alternativa.

[169] De acordo com Rumsey (2010), um *p-value* de 0.05 e um p-value de 0.01 indicam, respectivamente, que em 95% e 99% das vezes os resultados da amostra poderão se repetir caso o experimento seja realizado novamente com outras amostras aleatórias da mesma população sob as mesmas condições. Para Rumsey (2010), outros valores podem ser assumidos para o *p-value* e essa determinação depende de cada pesquisador.

Segundo a referida autora, se o *p-value* é maior ou igual a α, a hipótese nula deve ser aceita e, se o *p-value* é menor que α, a hipótese nula deve ser rejeitada. Em outras palavras, o resultado de um teste de hipótese é estatisticamente significativo quando, a partir do seu *p-value*, é possível rejeitar a hipótese nula devido à improbabilidade de que ela ocorra.

A consequência da rejeição da hipótese nula leva-nos a acreditar que a hipótese alternativa pode ser verdadeira. Levando em consideração os conceitos apresentados, definimos que o *p-value* do nosso teste seria de 0.05 por julgarmos esse nível de significância bastante aceitável para o propósito da nossa pesquisa.

4 RESULTADOS

Neste capítulo, apresentamos os resultados do trabalho desenvolvido nesta pesquisa. Inicialmente, no tópico 4.1, descrevemos e interpretamos os resultados do experimento. Em um segundo momento, no tópico 4.2, discutimos os resultados, procurando justificá-los em consonância com a perspectiva do trabalho de construção manual de *corpora* oportunizada pelo *ToGatherUp* e, ainda, esclarecendo o porquê de, em alguns casos, sermos favoráveis à escolha de *corpora* construídos de forma automática em detrimento de *corpora* elaborados manualmente. No tópico 4.2, também, destacamos os aspectos não quantificáveis que, embora não tenham sido contemplados na pesquisa, a nosso olhar, favorecem a utilização do *ToGatherUp*.

4.1 Interpretação e análise dos resultados do *T-Test*

Após o levantamento da amostra, a definição das hipóteses do teste e do *p-value*, fizemos o *T-Test* no SSPS, que gerou um relatório composto pelas Tabelas 2, 3 e 4 explicadas e analisadas nesta seção.

Tabela 2 – *Paired Samples Statistics*

		Mean	*N*	*Std. Deviation*	*Std. ErrorMean*
Pair1	ETCT – Método 1	1754,48	50	3027,984	428,222
	ETCT – Método 2	1623,34	50	3028,210	428,253

Fonte: SSPS.

A Tabela 2 exibe dados descritivos da comparação entre o par (*Pair 1*) de variáveis ETCT – Método 1 e ETCT – Método 2 do teste. A coluna *Mean* apresenta as médias de cada uma das variáveis e, conforme esperávamos, mostra que o ETCT da amostra construída com Método 2 (1623 segundos) é menor que o ETCT da amostra elaborada com o Método 1 (1754 segundos). A coluna *N* mostra a quantidade de registros de cada uma das amostras do teste (cinquenta textos). A coluna *Std. Deviation* evidencia o desvio padrão[170] e a coluna *Std. Error Mean* alude à média do desvio padrão das duas amostras.

[170] O desvio padrão "é a medida mais usada na comparação de diferenças entre conjuntos de dados, por ter grande precisão. O desvio padrão determina a dispersão dos valores em relação à média" (CORREIA, 2003, p. 61).

Apesar de a Tabela 2 apontar para uma redução do ETP para o Método 2, seus dados não são suficientes para determinarmos a rejeição da hipótese nula da pesquisa. Portanto, passamos à análise das demais tabelas. A seguir, expomos a Tabela 3:

Tabela 3 – *Paired Samples Correlations*

		N	*Correlation*	*Sig.*
Pair 1	ETCT – Método 1 & ETCT – Método 2	50	1,000	0,000

Fonte: SSPS

A coluna *Correlation* da Tabela 3 mostra o coeficiente de correlação (1,000) entre as variáveis ETCT – Método 1 e ETCT – Método 2 do teste. De acordo com Urdan (2011), a correlação é a força ou magnitude da relação entre duas variáveis e pode oscilar entre –1.00 e +1.00, sendo que um valor igual a 0.00 indica que não há uma relação significativa entre elas. Urdan (2011) explica, ainda, que, quanto mais próxima for a correlação dos valores –1.00 ou +1.00, mais forte é a relação entre as duas variáveis. Portanto, com base na Tabela 3, podemos considerar que existe uma relação forte e significativa entre o ETCT – Método 1 e o ETCT – Método 2. Na sequência, apresentamos a Tabela 4:

Tabela 4 – *Paired Samples Test*

		Paired Differences					*t*	*df*	*Sig. (2-tailed)*
		Mean	*Std. Deviation*	*Std. Error Mean*	*95% Confidence Interval of the Difference*				
					Lower	*Upper*			
Pair 1	ETCT 1 & ETCT 2	131,140	4,333	613	129,909	132,371	214,003	49	0,000

Fonte: SSPS.

Observações:
ETCT 1: ETCT – Método 1
ETCT 2: ETCT – Método 2

A Tabela 4 contém os dados mais importantes do *T-Test* da nossa pesquisa. Conforme podemos observar, a coluna *Mean* indica que, em média, o ETCT da amostra construída com

Método 1 é 131 segundos maior do que o ETCT da amostra elaborada com o Método 2. A coluna *Sig. (2-tailed)* traz o *p-value* do teste, que é igual a 0,000, um valor bem inferior ao *p-value* (0.05) estabelecido por nós para a garantia da significância estatística do *T-Test*.

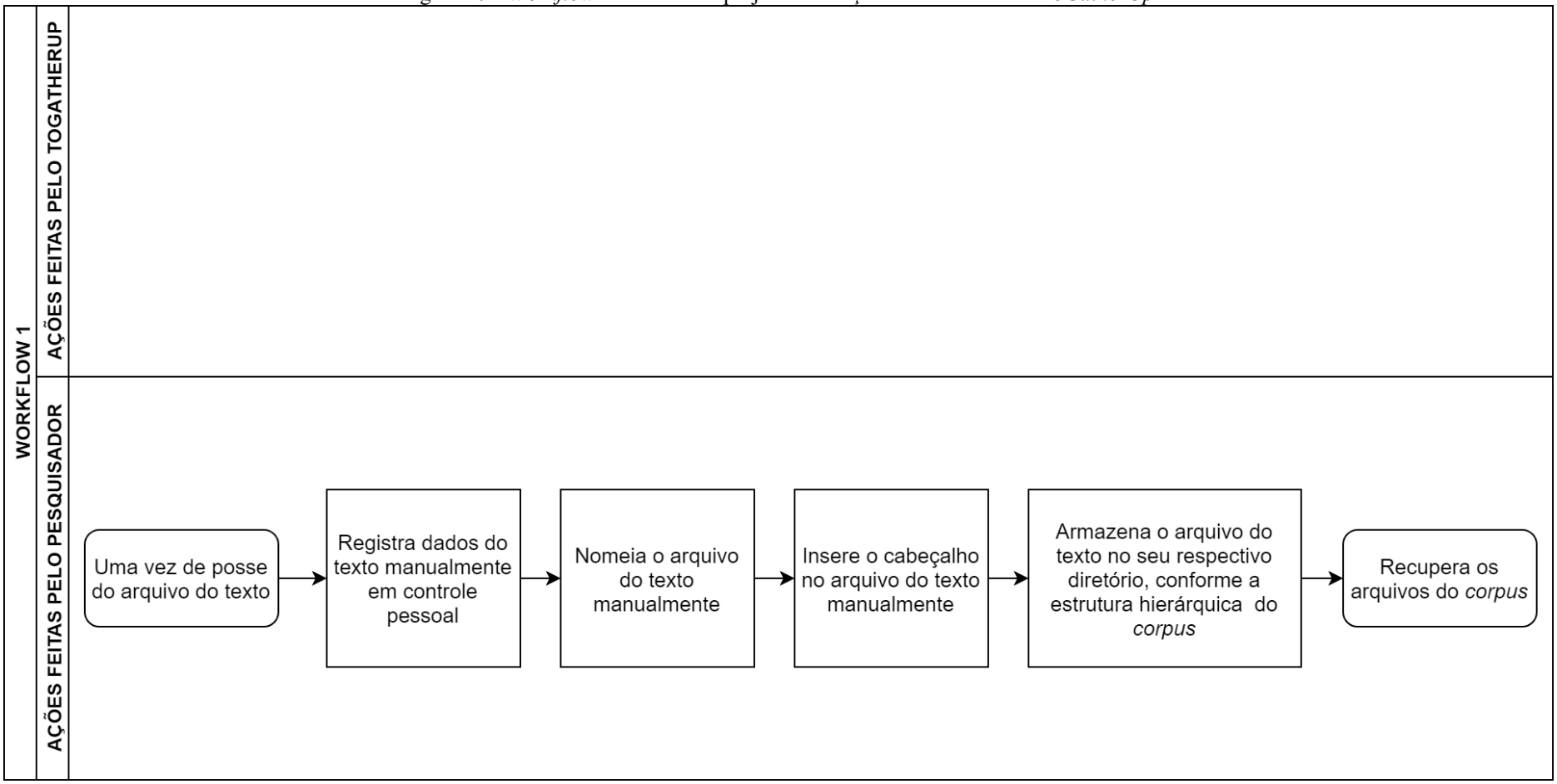
Portanto, com base nos dados apresentados, podemos rejeitar a hipótese nula da pesquisa (Hipótese nula (H_0): ETCT – Método 2 = ou > ETCT – Método 1) e afirmar, por inferência, que os resultados encontrados sugerem[171] que a incorporação do *ToGatherUp* reduz o ETP de construção manual de *corpora*.

4.2 Discussão dos resultados

A redução do ETP promovida pela incorporação do *ToGatherUp* em um projeto de construção manual de *corpora* pode ser explicada pela automatização de tarefas (cf. tópico 3.2 O *ToGatherUp* do terceiro capítulo) realizada pela ferramenta. A fim de compreendermos como a automatização de tarefas contribui para a redução do ETP, podemos comparar o fluxo de trabalho (*workflow*) dos projetos de construção do CoCLI. A Figura 40 exibe o *workflow* 1, que diz respeito ao projeto de construção do CoCLI em que não houve a incorporação do *ToGatherUp*, e a Figura 41 ilustra o *workflow* 2, que remete ao projeto em que houve a intervenção da ferramenta.

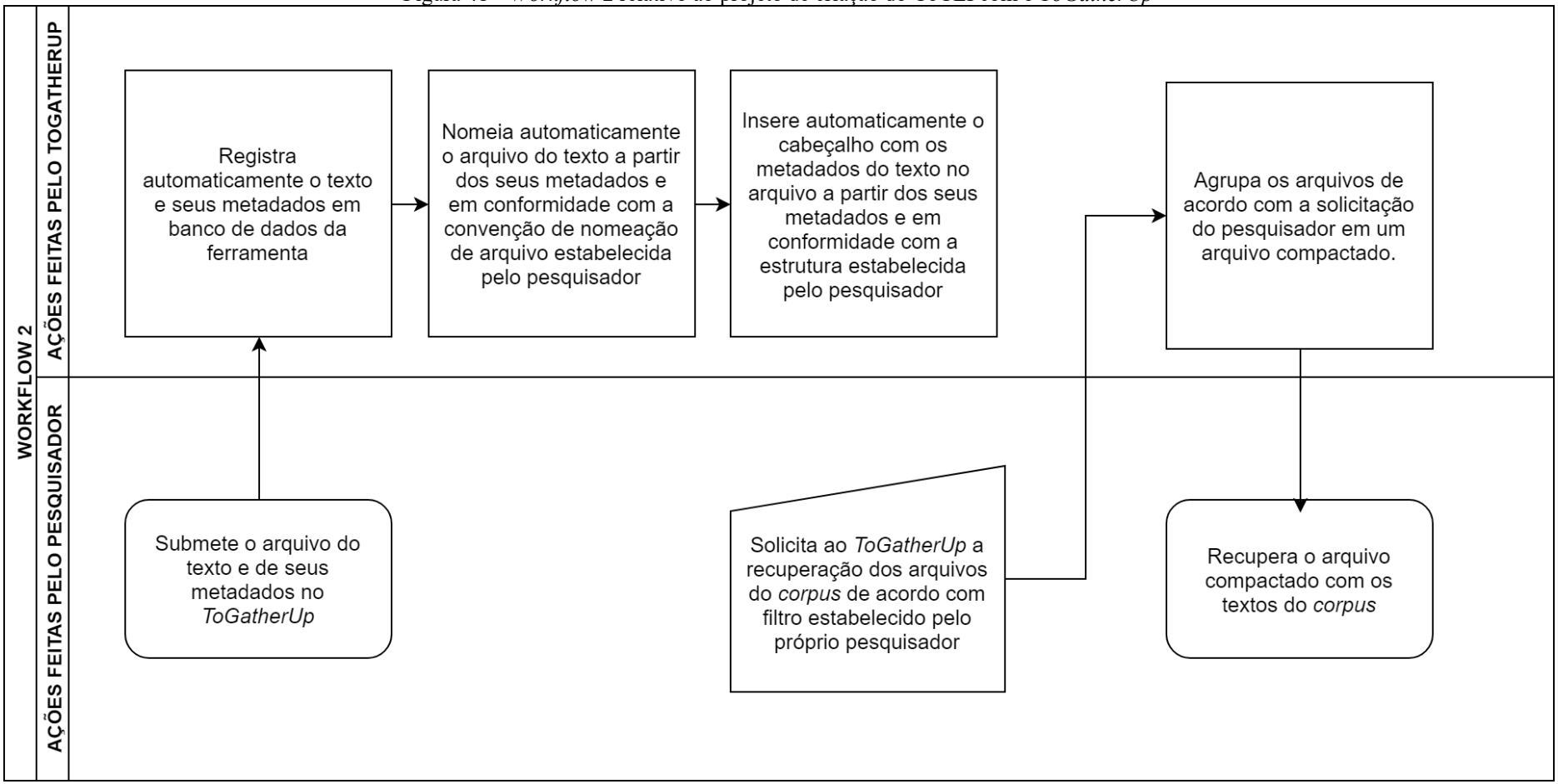
[171] Em média, o ETP de construção do CoCLI com a incorporação do *ToGatherUp* (*Mean* = 1623 *seconds*, *Std. Error Mean* = 428) foi menor que o ETP de elaboração do mesmo *corpus* sem a intervenção da ferramenta (*Mean* = 1754 *seconds*, *Std. Error Mean* = 428), t (49) = 214, $p < 0{,}05$.

Figura 40 – *Workflow*1 referente ao projeto de criação do CoCLI sem o *ToGatherUp*



Fonte: o autor.

Figura 41 – *Workflow* 2 relativo ao projeto de criação do CoCLI com o *ToGatherUp*



Fonte: o autor.

A parte inferior de ambas as Figuras mostra as tarefas que precisam ser executadas pelo pesquisador durante o projeto. Ao compararmos, podemos perceber que no *workflow* 1 todas as tarefas ficam a cargo do pesquisador e que no *workflow* 2 elas são distribuídas entre o pesquisador e o *ToGatherUp*. As tarefas realizadas pela ferramenta ocorrem de forma automática, o que praticamente diminui os EAs e, consequentemente, o ETP.

O *ToGatherUp* reduziu o ETP de construção do CoCLI, em média, 7,47%. Para o cálculo desse percentual, criamos um indicador que nomeamos de *ToGatherUp Effort Reduction Factor* (*T-Factor*). Ele informa o percentual da redução de esforço resultante do uso do *ToGatherUp* para a coleta de um texto, para realização de uma atividade (ou grupos de atividades) ou para a completude de um projeto. O valor do *T-Factor* é calculado pela fórmula: *T-Factor* = (Esforço[172] sem o *ToGatherUp* - Esforço com o *ToGatherUp*) / Esforço sem o *ToGatherUp*) * 100). A aplicação do modelo de cálculo do *T-Factor* para a amostra do nosso teste produziu o resultado da Figura 42.

Figura 42 – *T-Factor* da amostra do CoCLI

$$T\text{-}Factor = \frac{(\text{ETCT Método 1} - \text{ETCT Método 2})}{\text{ETCT Método 1}}$$

$$T\text{-}Factor = \frac{(87724 - 81167)}{87724}$$

$$T\text{-}Factor = 0{,}074 * 100$$

$$\textbf{\textit{T-Factor}} = \textbf{7,47\%}$$

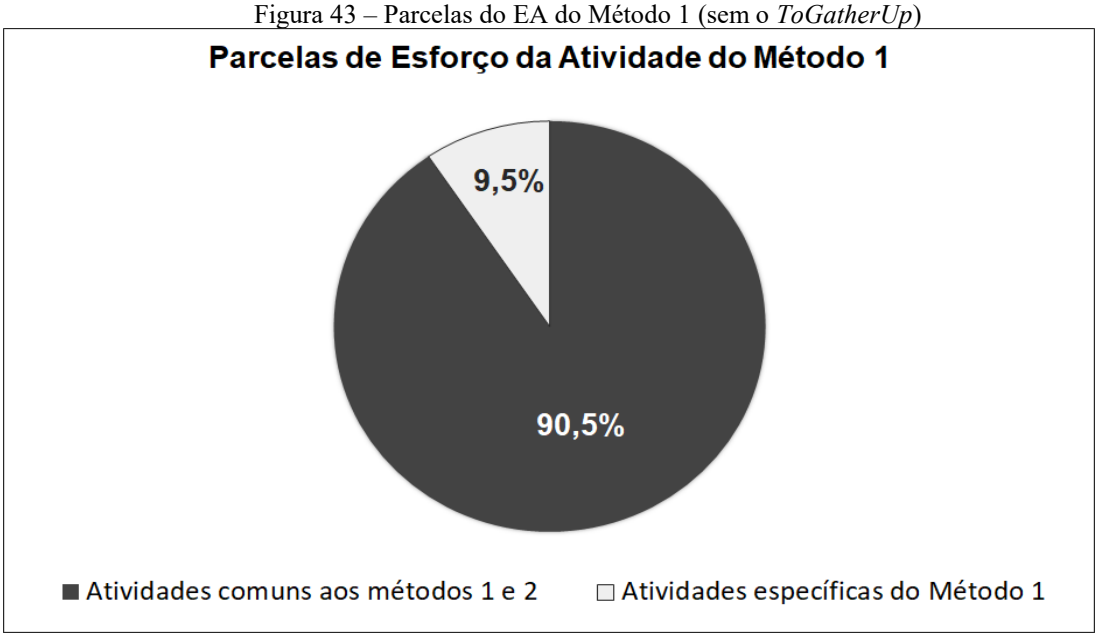
Fonte: o autor.

Com o propósito de melhorarmos a compreensão a respeito do efeito da incorporação do *ToGatherUp* no projeto de construção do CoCLI, a seguir, apresentamos os gráficos das Figuras 43 e 44 que, ao serem confrontados, permitem a

[172] O esforço pode ser o EA, o ETCT ou o ETP.

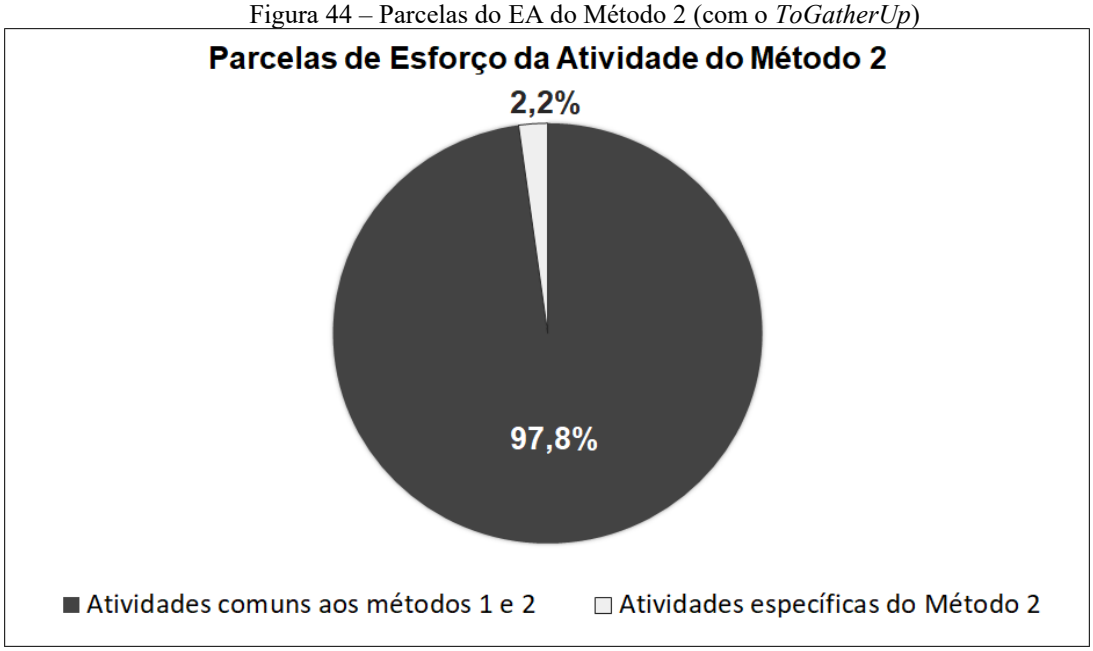
visualização da diferença percentual entre as parcelas de EAs específicas dos métodos 1 e 2 de construção do CoCLI.

Figura 43 – Parcelas do EA do Método 1 (sem o *ToGatherUp*)



Fonte: o autor.

Figura 44 – Parcelas do EA do Método 2 (com o *ToGatherUp*)



Fonte: o autor.

Podemos observar que o percentual do EA das parcelas das atividades específicas do Método 2 reduziu cerca de 7,3% em relação ao EA das parcelas das atividades específicas do Método 1. Por mais que o percentual apresentado esteja de acordo com o resultado *T-Factor* (7,47%) e com o resultado do *T-Test*, ele pode ser considerado baixo e provocar uma avaliação neutra ou negativa sobre a incorporação do *ToGatherUp* em

um projeto de construção de *corpus*. Todavia, do nosso ponto de vista, existem aspectos não quantificáveis acerca da incorporação do *ToGatherUp* em projetos de construção de *corpora* que podem propiciar uma avaliação positiva da ferramenta. No Quadro 13, apresentamos um apanhado de aspectos não quantificáveis que, a nosso olhar, contribuem para que o uso do *ToGatherUp* seja visto como uma boa opção para os pesquisadores.

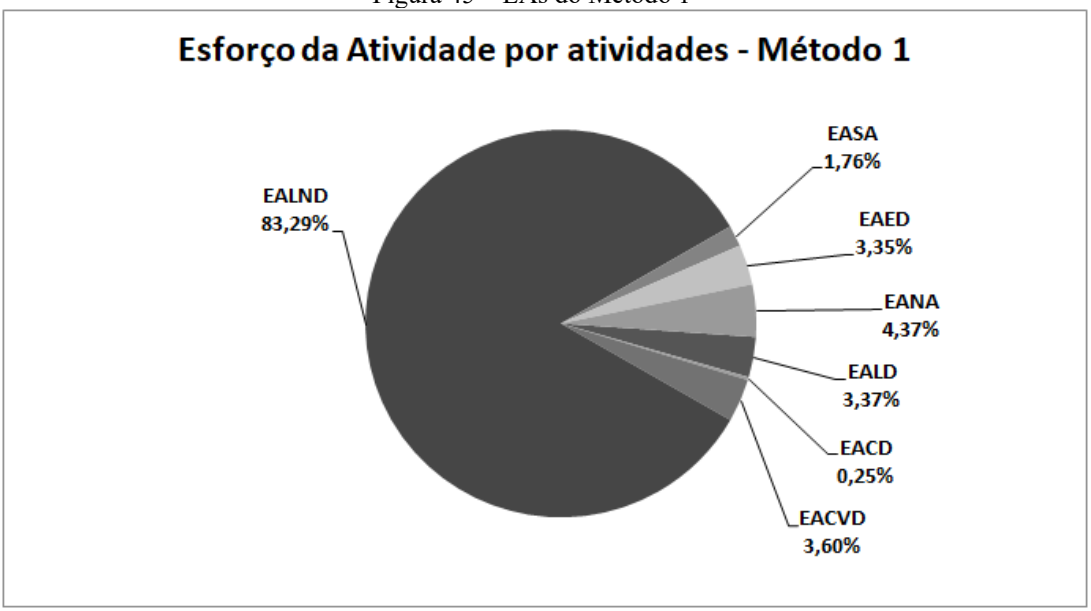
Quadro 13 – Aspectos positivos e não quantificáveis do uso do *ToGatherUp*

Aspectos não quantificáveis	**Vantagens proporcionadas ao pesquisador**
Automatização de procedimentos	● Redução de erros ocasionados pela intervenção humana. ● Padronização organizacional dos arquivos do *corpus*, diminuindo tarefas repetitivas. ● Promoção de maior confiabilidade ao projeto.
Natureza *on-line* e uso simultâneo por mais de um usuário no mesmo projeto	● Construção dos *corpora* de forma colaborativa independentemente do lugar em que esteja situado cada um dos componentes de um grupo de pesquisadores.
Restrição de acesso	● Acessibilidade restrita à interface do projeto e aos arquivos do *corpu*s. Somente os usuários cadastrados no sistema, com base em credenciais fornecidas pela coordenação do projeto, podem acessá-lo.
Segurança dos dados	● Redução do risco relacionado à perda de arquivos quando o pesquisador se vale apenas de HD de PCs para armazenar os *corpora*.

Fonte: o autor.

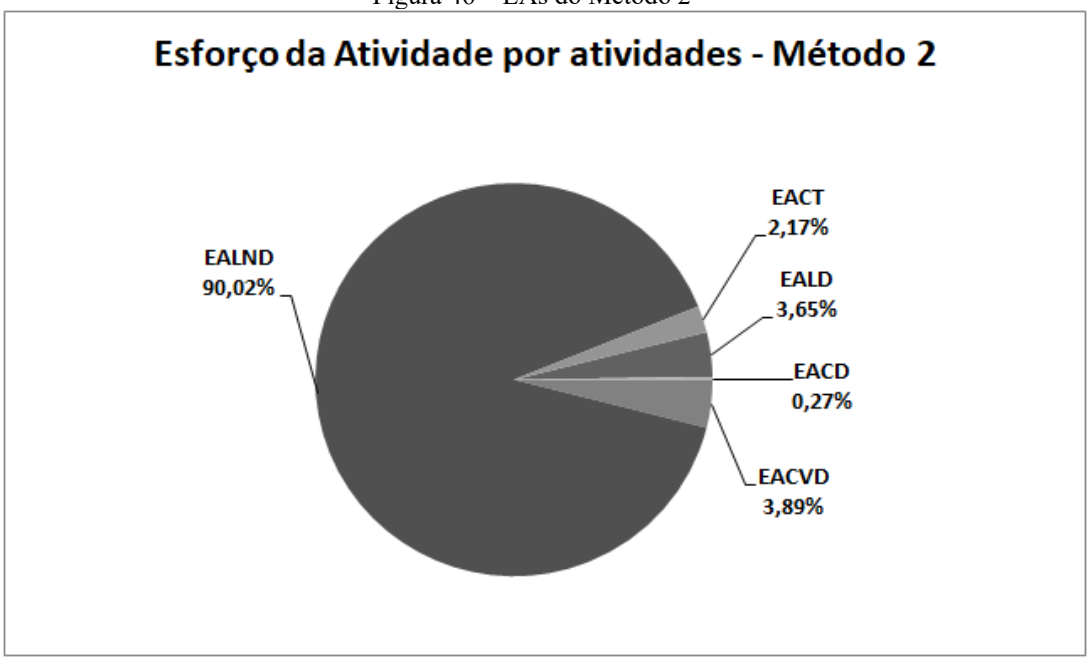
Apesar de não ser um dos objetivos desta pesquisa, a observação dos percentuais de EAs dos métodos 1 e 2 nos permitiu identificar um fato que nos chamou a atenção: o percentual correspondente ao EALND, em ambos os métodos de construção do CoCLI, foi superior aos demais esforços, atingindo 83,29% no Método 1 e 90,02% no Método 2. Esses percentuais podem ser observados nos gráficos das Figuras 45 e 46.

Figura 45 – EAs do Método 1[173]



Fonte: o autor.

Figura 46 – EAs do Método 2[174]



Fonte: o autor.

Os gráficos das Figuras 45 e 46 mostram que os percentuais do EALND são maiores do que todos os demais esforços somados juntos. Essa informação corrobora a ideia de Dasu e Johnson (2003) de que a limpeza e a normalização podem ocupar cerca

[173] A construção do gráfico não contemplou o EAOPD por ele corresponder a zero. O EAOPD assumiu esse valor por não termos executado a atividade de obtenção da permissão de uso dos textos do CoCLI, uma vez que não temos a intenção de publicizá-lo, conforme explicamos anteriormente.

[174] A construção do gráfico não contemplou o EAOPD devido às razões explicitadas na nota anterior. Também, não foram contemplados o EASA, o EAED e o EANA por corresponderem à zero. Essas três últimas métricas assumiram valor zero por terem sido automatizadas pelo *ToGatherUp*.

de 80% do tempo compreendido entre a obtenção de um texto e sua análise (cf. o tópico 2.5 O tempo e o esforço na construção de um *corpus* do segundo capítulo), além de ajudar a entender o porquê de, em alguns casos, sermos favoráveis à escolha de *corpora* construídos de forma automática em detrimento de *corpora* elaborados manualmente.

Além disso, a grande dimensão dos percentuais do EALND pode ser considerada um indicador das possíveis complicações do uso de *web corpora* nas pesquisas em que existe a preocupação quanto à precisão de análises, visto que as ferramentas de coleta automática de textos, no estágio atual da tecnologia, não conseguem lidar com os problemas apontados no tópico 2.5 O tempo e o esforço na construção de um *corpus* no segundo capítulo desta pesquisa.

A possível incorporação do *ToGatherUp* em uma pesquisa não resolve as dificuldades que aludem à limpeza e à normalização de textos. Em virtude disso e do negligenciamento da atividade de limpeza e normalização de textos por parte das ferramentas de construção automática de *corpora*, sugerimos que pesquisas futuras invistam em soluções que possam melhorar a confiabilidade dos resultados referentes às análises de *web corpora*.

5 CONSIDERAÇÕES FINAIS

A presente pesquisa é o resultado de um trabalho sistemático para a determinação do efeito da incorporação do *ToGatherUp* em projetos de construção manual de *corpora*. A busca por esse objetivo guiou o desenvolvimento deste trabalho de modo bastante produtivo ao possibilitar que: a) identificássemos, por meio de um levantamento, a lacuna referente à inexistência de ferramentas atuais da LC que oferecem suporte às atividades de elaboração manual de *corpora*; b) propuséssemos uma sistematização do trabalho de criação manual de *corpora*, envolvendo princípios e métodos da LC e da área de Gerenciamento de Projetos; c) desenvolvêssemos métricas e um método de mensurar o esforço dos projetos de construção manual de *corpora*.

Até onde pudemos verificar por meio da revisão bibliográfica da LC, nossa Dissertação de Mestrado é a primeira a propor uma forma de mensurar o esforço necessário para a realização de projetos de elaboração manual de *corpora* e a avaliar o efeito da incorporação de uma ferramenta computacional de suporte para tais projetos. Além das contribuições citadas, o *ToGatherUp*, mesmo tendo sido desenhado especialmente para a realização desta pesquisa, possui potencial para ser aplicado em outros projetos após as implementações[175] do Quadro 14.

[175] Configuramos a versão inicial do *ToGatherUp* em consonância com o escopo da nossa pesquisa e, devido a restrições de tempo, não disponibilizamos a outros pesquisadores a opção de configurá-lo conforme as especificações de seus projetos. Na verdade, para que o *ToGatherUp* possa ser utilizado pelo público, precisamos promover as melhorias apresentadas no Quadro 14, que devem ser implementadas nas próximas versões do sistema.

Quadro 14 – Implementações necessárias para disponibilização pública do *ToGatherUp*

Funcionalidades	Estado atual	Melhorias
Cadastro de Textos	Ausência de interface para customização dos campos do formulário do Cadastro de Textos. Os campos do formulário do Cadastro de Textos da nossa pesquisa foram definidos diretamente no banco de dados do sistema.	Criar interface para customização, por parte do usuário, conforme critérios da pesquisa dele, dos campos do formulário do Cadastro de Textos.
Árvore de Domínio	Ausência de interface para criação da Árvore de Domínio. A Árvore de Domínio da Computação usada em nossa pesquisa foi definida diretamente no banco de dados do sistema.	Criar interface para elaboração da Árvore de Domínio ou outra forma hierárquica de organização desejada pelo usuário, conforme critérios da pesquisa dele.
Interoperabilidade	O *ToGatherUp* aceita somente arquivos no formato TXT. Os arquivos do *corpus* são exportados somente no formato TXT.	Permitir a inclusão de textos em outros formatos, como PDF ou XML.
Edição e remoção de textos	O *ToGatherUp* não possui uma interface para a realização de edições ou exclusões dos textos do *corpus*. Tais procedimentos precisam ser realizados manualmente no banco de dados do sistema e no diretório onde o arquivo foi alocado.	Incluir uma interface com editor de texto para a edição dos arquivos do *corpus* dentro do *ToGatherUp*. Acrescentar ao Gerenciador de *corpus* a opção para que o texto seja excluído.
Tutorial	O *ToGatherUp* não possui tutorial.	Criar tutorial de uso do sistema.

Fonte: o autor.

As mudanças citadas no Quadro 14 permitirão que pesquisadores possam configurar o *ToGatherUp* para uso em seus projetos. Por essa razão, em nossa futura pesquisa de Doutorado, pretendemos implementá-las e lançar a ferramenta,

gratuitamente, para que possa ser utilizada pela comunidade acadêmica. Além disso, intencionamos ampliar o leque de funcionalidades do *ToGatherUp* com a inclusão de recursos de análise de *corpora* e de recursos para a solução de problemas relacionados à atividade de limpeza e normalização de textos que, conforme o trabalho atual mostra, constitui-se como um dos gargalhos dos projetos de produção de *corpora*. Pretendemos, ainda, envolver pesquisadores, profissionais e estudantes no melhoramento do *ToGatherUp*, convidando-os para testá-lo e incentivando-os a registrar suas avaliações sobre o seu uso.

Conforme exposto no tópico 1.2 Contexto da pesquisa, presente no primeiro capítulo, nosso trabalho partiu de um problema concreto que também pode ocorrer em outras investigações científicas. Nesse sentido, acreditamos que os métodos, os recursos e as discussões que apresentamos nesta pesquisa possam ser de grande valia para aqueles que pretendem embarcar em um projeto de construção manual de *corpora.*

REFERÊNCIAS

ALMEIDA, M. B. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. **Ciência da informação**, Brasília, v. 31, n. 2, p. 5-13, maio/ago. 2002. DOI: https://doi.org/10.1590/S0100-19652002000200001. Disponível em: http://www.scielo.br/pdf/ci/v31n2/12903. Acesso em: 2 abr. 2019.

ALUÍSIO, S. M.; ALMEIDA, G. M. B. O que é e como se constrói um corpus? Lições aprendidas na compilação de vários *corpora* para pesquisa linguística. **Calidoscópio**, São Leopoldo, v. 4, n. 3, p. 156-178, set./dez. 2006. Disponível em: http://revistas.unisinos.br/index.php/calidoscopio/article/view/6002. Acesso em: 2 abr. 2019.

ALVES, R. C. V. **Metadados como elementos do processo de catalogação**. Orientadora: Dra. Plácida Leopoldina Ventura Amorim da Costa Santos. 2010. 132 f. Tese (Doutorado em Ciência da Informação) – Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, Marília, 2010. Disponível em: <https://repositorio.unesp.br/handle/11449/103361>. Acesso em: 2 abr. 2019.

ANTHONY, L. A critical look at software tools in corpus linguistics. **Linguistic Research**, Dongdaemun-gu, Seou, v. 30, n. 2, p. 141-161, 2013. DOI: https://doi.org/10.17250/khisli.30.2.201308.001. Disponível em: https://www.laurenceanthony.net/research/20130827_linguistic_research_paper/linguistic_research_paper_final.pdf. Acesso em: 5 dez. 2018.

ANTHONY, L. AntFileSplitter. Version 1.0.0. [Computer Software]. Tokyo: Waseda University, 2015. Disponível em: http://www.laurenceanthony.net. Acesso em: 5 dez. 2018.

ANTHONY, L. EncodeAnt. Version 1.2.0. [Computer Software]. Tokyo: Waseda University, 2016. Disponível em: http://www.laurenceanthony.net. Acesso em: 2 abr. 2019.

ANTHONY, L. AntCorGen. Version 1.1.1. [Computer Software]. Tokyo: Waseda University, 2018. Disponível em: http://www.laurenceanthony.net/software. Acesso em: 5 dez. 2018.

ANTHONY, L. AntConc. Version 3.5.8. [Computer Software]. Tokyo: Waseda University, 2019. Disponível em: http://www.laurenceanthony.net/software/antconc/. Acesso em: 6 mar. 2019.

ATKINS, S.; CLEAR, J.; OSTLER, N. Corpus design criteria. **Literary and linguistic computing**, Oxford, Oxford University Press, v. 7, n. 1, p. 1-16, jan. 1992. DOI: https://doi.org/10.1093/llc/7.1.1. Disponível em: https://academic.oup.com/dsh/article-abstract/7/1/1/1028498?redirectedFrom=fulltext. Acesso em: 17 abr. 2019.

BAKER, P. Corpus Methods in Linguistics. *In*: LITOSSELITI, L. (ed.). **Research methods in linguistics.** New York: Continnum International Publishing Group, 2010. p. 93-113.

BAKER, P.; HARDIE, A.; MCENERY, T. **A glossary of corpus linguistics**. Edinburgh: Edinburgh University Press, 2006.

BARBOSA, M. A. Contribuição ao estudo de aspectos da tipologia de obras lexicográficas. **Ciência da informação**, Brasília, v. 24, n. 3, 1995. Disponível em: http://revista.ibict.br/ciinf/article/view/572/573. Acesso em: 22 jun. 2018.

BARONI, M.; BERNARDINI, S. BootCaT. Version 1.08. [Computer Software]. Trento/Forlì: Universities of Bologna, 2004. Disponível em: http://bootcat.dipintra.it. Acesso em: 2 abr. 2019.

BARONI, M. *et al*. WebBootCaT: a web tool for instant corpora. *In*: 12th EURALEX INTERNATIONAL CONGRESS, 2006. **Proceedings [...]** Torino: Edizioni dell'Orso s.r.l., 2006. p. 123-131. Disponível em: https://euralex.org/publications/webbootcat-a-web-tool-for-instant-corpora/. Acesso em: 2 abr. 2019.

BARREAU, D.; NARDI, B. Finding and reminding: File organization from the desktop. **ACM SIGCHI Bulletin**, New York, v. 27, n. 3, p. 39-43, jul.1995. DOI: https://doi.org/10.1145/221296.221307. Disponível em: https://dl.acm.org/citation.cfm?id=221307. Acesso em: 17 abr. 2019.

BATEMAN, J. **Multimodality and genre**: A foundation for the systematic analysis of multimodal documents. United Kingdom: Palgrave Macmillan, 2008. DOI: https://doi.org/10.1057/9780230582323.

BERBER SARDINHA, T. A influência do tamanho do corpus de referência na obtenção de palavras chave. **DIRECT Papers 38**, São Paulo/Liverpool: LAEL PUCSP, p. 1-18, 1999. Disponível em: http://www2.lael.pucsp.br/direct/DirectPapers38.pdf. Acesso em: 22 jun. 2018.

BERBER SARDINHA, T. **Linguística de *Corpus***. São Paulo: Manole, 2004. 410 p.

BERGH, G.; ZANCHETTA, E. (2008). Web linguistics. *In*: LÜDELING, A.; KYTÖ, M. (ed.). **Corpus Linguistics**: An International Handbook. Berlin: Mouton de Gruyter, 2008. p. 309-327.

Beyond VBA Tutorial. **Word Macro**: Deleting All Headers and Footers from an Active Document. 2019. Disponível em: http://vba.relief.jp/word-macro-delete-all-headers-and-footers-active-document/. Acesso em: 20 fev. 2019.

BIANCHI, F. **Culture, Corpora and Semantics**: methodological issues in using elicited and corpus data for cultural comparison. Lecce: ESE Salento University Publishing, 2012. Disponível em: http://siba-ese.unisalento.it/index.php/culturecorpora/article/viewFile/12427/11066. Acesso em: 10 jan. 2019.

BIBER, D. Representativeness in Corpus Design. **Literary and Linguistic Computing**, Oxford, v. 8, n. 4, p. 223-257, out. 1993. DOI: https://doi.org/10.1093/llc/8.4.243. Disponível em: http://otipl.philol.msu.ru/media/biber930.pdf. Acesso em: 2 abr. 2019.

BLECHA, J. **Building Specialized Corpora**. Supervisor: PhDr. Jarmila Fictumová. 2012. 159 f. Master's Diploma Thesis (English Language and Literature) – Faculty of Arts, Department of English and American Studies, Masaryk University, Brno, 2012. Disponível em: https://is.muni.cz/th/aki90/179991_Building_Specialized_Corpora.pdf. Acesso em: 2 abr. 2019.

BONONNO, R. Terminology for translators: an implementation of ISO 12620. **Meta**, Montréal, v. 45, n. 4, p. 646-669, dez. 2000. DOI: https://doi.org/10.7202/002101ar. Disponível em: https://www.erudit.org/fr/revues/meta/2000-v45-n4meta161/002101ar.pdf. Acesso em: 22 jun. 2018.

BOWKER, L.; PEARSON, J. **Working with specialized language**: a practical guide to using corpora. London: Routledge, 2002. DOI: https://doi.org/10.4324/9780203469255.

BRASE, C. H.; BRASE, C. P. **Understandable Statistics**: Concepts and Methods, 10. ed., Boston: Cengage Learning, 2011.

BRASIL. Lei nº. 9.610, de 19 de fevereiro de 1998. Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências. **Diário Oficial da União**, Brasília, 20 fev. 1998. Seção 1.

BRASIL. Resolução Nº 5, de 16 de novembro de 2016. Institui as Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação, abrangendo os cursos de bacharelado em Ciência da Computação, em Sistemas de Informação, em Engenharia de Computação, em Engenharia de Software e de licenciatura em Computação, e dá outras providências. **Diário Oficial da União**, Brasília, n. 220, 17 nov. 2016. Seção 1, p. 22-24.

CABRÉ, M. T. **Terminology**: Theory, Methods and Applications. Tradução Anne DeCesaris. Amsterdam/ Philadelphia: John Benjamin Publishing, 1999. DOI: https://doi.org/10.1075/tlrp.1

CABRÉ, M. T. Theories of terminology: their description, prescription and explanation. **Terminology**: **International Journal of Theoretical and Applied Issues in specialized communication**, Amsterdam, v. 9, n. 2, p. 163-200, 2003. DOI: https://doi.org/10.1075/term.9.2.03cab. Disponível em: https://www.jbe-platform.com/content/journals/10.1075/term.9.2.03cab. Acesso em: 17 abr. 2019.

CARDOSO. S. A. F. **TermosTeo**: a elaboração de vocabulários monolíngues de termos da Teologia em um estudo conduzido por *corpus*. Orientador: Dr. Guilherme Fromm. 2017. 340 f. Tese (Doutorado em Estudos Linguísticos) – Instituto de Letras e Linguística, Universidade Federal de Uberlândia, Uberlândia, 2017. Disponível em: https://repositorio.ufu.br/handle/123456789/21349?mode=full. Acesso em: 15 abr. 2019.

CODD, E.F. A relational model of data for large shared data banks. **Communications of the ACM**, Philadelphia, v. 13, n. 6, p. 377-387, 1970. DOI: https://doi.org/10.1145/362384.362685. Disponível em: https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf. Acesso em: 2 abr. 2019.

CONNOR, U; UPTON, T. Introduction. *In*: CONNOR, U; UPTON, T. (org.). **Discourse in the professions:** Perspectives from corpus linguistics. Amsterdam/Philadelphia: John Benjamins, 2004. p. 1-8. DOI: https://doi.org/10.1075/scl.16.01con.

COOPER, W. A. L. J. *et al*. **Method and system for labeling a document for storage, manipulation, and retrieval**. US Pat. US5448375A, 5 set. 1995. 21 p. Disponível em: https://patents.google.com/patent/US5448375A/en. Acesso em: 17 abr. 2019.

CORREIA, M. S. B. B. **Probabilidade e estatística**. 2. ed. Belo Horizonte: PUC Minas Virtual, 2003. 116 p. Disponível em: <http://estpoli.pbworks.com/f/livro_probabilidade_estatistica_2a_ed.pdf>. Acesso em: 25 fev. 2019.

DASU, T.; JOHNSON, T. **Exploratory data mining and data cleaning**. Hoboken: John Wiley & Sons, 2003. DOI: https://doi.org/10.1002/0471448354.

DAVIES, M. Brigham. English-Corpora.org. Brigham Young University (BYU). Disponível em: https://www.english-corpora.org. Acesso em: 28 mar. 2019.

DIETRICH, D. *et al*. **Open data handbook**. Open Knowledge International, 2009. Disponível em: http://opendatahandbook.org/. Acesso em: 17 abr. 2019.

DODGE, Y. **The concise encyclopedia of statistics**. New York: Springer-Verlag, 2008.

DOURISH, P. The appropriation of interactive technologies: Some lessons from placeless documents. **Computer Supported Cooperative Work (CSCW)**, Dordrecht, v. 12, n. 4, p. 465-490, 2003. DOI: https://doi.org/10.1023/A:1026149119426. Disponível em: https://link.springer.com/article/10.1023/A:1026149119426. Acesso em: 17 abr. 2019.

EDWARD, R. P. Computational tools and methods for corpus compilation and analysis. *In*: BIBER, D; REPPEN, R. **The Cambridge Handbook of English corpus linguistics**. Cambridge: Cambridge University Press, 2015. p. 32-49.

ESCARTÍN, C. P. Design and compilation of a specialized Spanish-German parallel corpus. *In*: LANGUAGE RESOURCES AND EVALUATION CONFERENCE (LREC), 2012, Istanbul. **Proceedings [**…] Istanbul: European Language Resources Association (ELRA), 2012. p. 2199-2206. Disponível em: http://www.lrec-conf.org/proceedings/lrec2012/pdf/577_Paper.pdf. Acesso em: 2 abr. 2019.

EVANS, D. Compiling a corpus. In: Introduction to Corpus investigative techniques: **an on-line information pack about corpus investigation techniques for the Humanities**. Birmingham: University of Birmingham, 2007. Disponível em: https://www.birmingham.ac.uk/research/activity/corpus/publications/introduction-corpus-investigative-techniques.aspx. Acesso em: 10 jan. 2019.

FENTON, N., BIEMAN, J. **Software Metrics**: A Rigorous and Practical Approach. 3. ed. Boca Raton: CRC Press, 2014. DOI: https://doi.org/10.1201/b17461.

FONSECA FILHO, C. **História da Computação**: o caminho do pensamento e da tecnologia. Porto Alegre: EdiPUCRS, 2007. 204 p. Disponível em: http://www.pucrs.br/edipucrs/online/historiadacomputacao.pdf. Acesso em: 17 abr. 2019.

FRANKENBERG-GARCIA, A. Prefácio. *In*: SHEPHERD, T. M. G.; BERBER SARDINHA, T.; PINTO, M. V. (org.). **Caminhos da linguística de corpus**. Mercado de Letras, 2012.

FROMM, G. **Proposta para um modelo de glossário de informática para tradutores**. Orientador: Dr. Francis Henrik Aubert. 2002. 82 f. Dissertação (Mestrado em Letras) – Faculdade de Filosofia, Letras e Ciências Humanas, Universidade de São Paulo, São Paulo, 2002. Disponível em: http://www.ileel.ufu.br/guifromm/wp-content/uploads/2014/05/dissertacao.pdf. Acesso em: 2 abr. 2019.

FROMM, G. O Uso de *Corpora* na análise linguística. **Revista Factus,** São Paulo, v. 1, n. 1, p.69-76, 2003. Disponível em: http://www.ileel.ufu.br/guifromm/upload/ousodecorporanaproducaolinguistica.pdf. Acesso em: 17 abr. 2019.

FROMM, G. A questão da taxonomia num *corpus* colaborativo para construção de um vocabulário na área de linguística. *In*: SIMPÓSIO INTERNACIONAL DE LETRAS E LINGUÍSTICA (SILEL), 2013, Uberlândia. **Anais [...]** Uberlândia: EDUFU, v. 3, n. 1. 2013. Não paginado. Disponível em: http://www.ileel.ufu.br/anaisdosilel/pt/. Acesso em: 17 abr. 2019.

FROMM, G. **VoTec**: a construção de vocabulários eletrônicos para aprendizes de tradução. Orientadora: Dra. Stella Esther Ortweiller Tagnin. 2007. 214 f. Tese (Doutorado em Letras) – Faculdade de Filosofia, Letras e Ciências Humanas, Universidade de São Paulo, São Paulo, 2007. Disponível em: http://www.teses.usp.br/teses/disponiveis/8/8147/tde-08072008-150855/pt-br.php. Acesso em: 2 abr. 2019.

FROMM, G.; VICTOR, S. UltraLex. Versão 1.1.1, 2018. Disponível em: http://ultralex.ileel.ufu.br. Acesso em: 5 dez. 2018.

GARRETSON, G. Desiderata for Linguistic Software Design. **Internatinal Journal of English Studies** (IJES), Murcia, v. 8, n. 1, 67-94, 2008. Disponível em: http://revistas.um.es/ijes/article/view/49101. Acesso em: 2 abr. 2019.

GOOGLE, 2019. **Refinar pesquisas na Web**. Disponível em: https://support.google.com/websearch/answer/2466433?hl=pt-BR. Acesso em: 1 abr. 2019.

GRAMA, D. F. **Uma análise lexicográfica dos elementos coesivos sequenciais do português para a elaboração de uma proposta de definição**: um estudo com base em corpus. Orientador: Dr. Guilherme Fromm. 2016. 371 f. Dissertação (Mestrado em Estudos Linguísticos) – Instituto de Letras e Linguística, Universidade Federal de Uberlândia, Uberlândia, 2016. Disponível em: https://repositorio.ufu.br/handle/123456789/18084. Acesso em: 15 abr. 2019.

GREFENSTETTE, G.; TAPANAINEN, P. What is a word, what is a sentence?: problems of Tokenisation. *In*: 3RD INTERNATIONAL CONFERENCE ON COMPUTATIONAL LEXICOGRAPHY (COMPLEX'94), 1994, Budapest. **Proceedings [...]** Budapest: Hungarian Academy of Sciences, 1994. p. 79-87. Disponível em: http://real-eod.mtak.hu/6835/. Acesso em: 17 abr. 2019.

GRIES, S. T. What is Corpus Linguistics?, **Language and Linguistics Compass**, Hoboken, v. 3, n. 5, p. 1225-1241, 2009. DOI: https://doi.org/10.1111/j.1749-818X.2009.00149.x. Disponível em: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1749-818X.2009.00149.x. Acesso em: 2 abr. 2019.

HANSEN-SCHIRRA, S. Linguistic enrichment and exploitation of the Translational English Corpus. *In*: CORPUS LINGUISTICS CONFERENCE 2003, Lancaster. **Proceedings [...]** Lancaster: Lancaster University, 2003. p. 288-297. Disponível em: http://ucrel.lancs.ac.uk/publications/CL2003/papers/hansen.pdf. Acesso em: 17 abr. 2019.

HARDIE, A. Corpus Linguistics and the Languages of South Asia: Some Current Research Directions. *In*: BAKER, P. (ed.). **Contemporary corpus linguistics**. London: Continuum, 2012. p. 262-288.

HART, M. S. Projeto Gutenberg, 1971. Disponível em: https://www.gutenberg.org/wiki/PT_Principal. Acesso em: 17 abr. 2019.

HOOVER, D. L.; CULPEPER, J.; O'HALLORAN, K. **Digital literary studies**: Corpus approaches to poetry, prose, and drama. New York: Routledge, 2014. DOI: https://doi.org/10.4324/9780203698914.

HÜNING, M. TextSTAT. Version 2.9. Berlin: Freie Universität, 2014. Disponível em: http://neon.niederlandistik.fu-berlin.de/de/textstat/. Acesso em: 25 jan. 2019.

KAPTELININ, V. Creating Computer-Based Work Environments: An Empirical Study of Macintosh Users. *In*: 1996 ACM SIGCPR/SIGMIS CONFERENCE ON COMPUTER PERSONNEL RESEARC, 1996, Denver. **Proceedings [...]** Denver: ACM, 1996, p. 360-366. DOI: https://doi.org/10.1145/238857.238921. Disponível em: https://dl.acm.org/citation.cfm?id=238921. Acesso em: 17 abr. 2019.

KEHOE, A.; GEE, M. New corpora from the web: making web text more 'text-like'. **Studies in Variation, Contacts and Change in English**, Helsinki, v. 2, 2007. Disponível em: http://www.helsinki.fi/varieng/series/volumes/02/kehoe_gee/. Acesso em: 18 jan. 2019.

KENNEDY, G. **An Introduction to Corpus Linguistics**. New York: Longman, 1998.

KILGARRIFF, A.; RYCHLÝ, P. Sketch Engine. Lorient: EURALEX, 2004. Disponível em: http://www.sketchengine.eu. Acesso em: 25 jan. 2019.

KLEIBER, I. TextDirectory. Heidelberg: Heidelberg University, 2018. Disponível em: https://github.com/IngoKl/textdirectory. Acesso em: 25 jan. 2019.

KLEIBER, I.; BERBERICH, K. Corpus Analysis. Heidelberg: Heidelberg University, 2018. Disponível em: https://corpus-analysis.com/. Acesso em: 25 jan. 2019.

KOESTER, A. Building small specialised corpora. *In*: O'KEEFFE, A.; MCCARTHY, M. J. (org.). **The Routledge handbook of corpus linguistics.** London**:** Routledge, 2010. p. 66-79.

KRIEGER, M. G.; FINATTO, M. J. B. **Introdução à terminologia**: teoria e prática. São Paulo: Contexto, 2004. 223p.

KÜBLER, N.; ASTON, G. Using Corpora in Translation. *In*: O'KEEFFE, A.; MCCARTHY, M. J. (org.). **The Routledge handbook of corpus linguistics.** London**:** Routledge, 2010. p. 501-515.

LEECH, G. Adding Linguistic Annotation. *In*: WYNNE, M. (ed.). **Developing linguistic corpora:** A guide to good practice. Oxford: Oxbow Books, 2005. p. 17-29. Disponível em: http://ota.ox.ac.uk/documents/creating/dlc/. Acesso em: 2 abr. 2019.

LEGGETT, E. **Digitization and digital archiving:** a practical guide for librarians. Lanham: Rowman & Littlefield Publishing, 2014. 226 p.

LIANG, M.; JIAJIN, X. Sub-corpus Creator. Beijing: Beijing Foreign Suties University, 2011. Disponível em: http://corpus.bfsu.edu.cn/tools. Acesso em: 5 dez. 2018.

GATHER UP. *In*: Macmillan Dictionary. 2018. Disponível em: https://www.macmillandictionary.com/dictionary/british/gather-up. Acesso em: 21 jun. 2018.

MACMULLEN , W. J. Requirements Definition and Design Criteria for Test Corpora in Information Science. **SILS Technical Report 2003-03**. School of Information and Library Science: University of North Carolina at Chapel Hill. p. 3-21, 2003. Disponível em: https://sils.unc.edu/sites/default/files/general/research/TR-2003-03.pdf. Acesso em: 10 jan. 2019.

MARTINET, A. **Elementos de linguística geral**. 8. ed. Lisboa: Martins Fontes, 1978.

MARTÍNEZ, G. E. S. **Introducción a los corpus lingüísticos**. Ciudad de México: Instituto de Ingeniería UNAM, 2017.

MCENERY, T.; HARDIE, A. **Corpus linguistics**: Method, theory and practice. Cambridge: Cambridge University Press, 2011. DOI: https://doi.org/10.1017/CBO9780511981395.

MCENERY, T.; XIAO, R.; TONO, Y. **Corpus-based language studies**: An advanced resource book. London/New York: Routledge, 2006. Disponível em: https://www.lancaster.ac.uk/fass/projects/corpus/ZJU/xCBLS/chapters/A10.pdf. Acesso em: 10 jan. 2019.

MEYER, C. F. **English corpus linguistics:** an introduction. Cambridge: Cambridge University Press, 2004.

MICHAEL, F. Introdução às Expressões Regulares. Tradução de Lúcia Ayako Kinoshita. São Paulo: Novatec Editora; Sebastopol, CA: O'Reilly, 2012.

MICROSOFT. Compactar e descompactar arquivos, 2016. Disponível em: https://support.microsoft.com/pt-br/help/14200/windows-compress-uncompress-zip-files. Acesso em: 16 de jan. 2019.

MINSHALL, D. E. **A Computer Science Word List**. Supervisor: Dr. Vivienne Rogers. 2013. 98 f. Master of Arts (MA TEFL) – University of Swansea, Swansea, 2013. Disponível em: https://www.baleap.org/wp-content/uploads/2016/03/Daniel-Minshall.pdf. Acesso em: 10 jan. 2019.

MORRISON, A., POPHAM, M.; WIKANDER, K. **Creating and Documenting Electronic Texts**. Disponível em: <http://ota.ox.ac.uk/documents/creating/cdet/chap3.html> Acesso em: 2 abr. 2019. Não paginado.

NAJORK, M. A.; HEYDON, C. A. **System and method for associating an extensible set of data with documents downloaded by a web crawler**. U.S. Patent n. 6,351,755, 26 fev. 2002.

NELSON, M. Building a written corpus: What are the basics? *In*: O'KEEFFE, A.; MCCARTHY, M. J. (org.). **The Routledge handbook of corpus linguistics.** London**:** Routledge, 2010. p. 53-65.

NETO, J. B. Apontamentos para uma tipologia dos modelos linguísticos. **Revista Letras**, Curitiba, v. 29, p. 75-87, 1980. Disponível em: <https://revistas.ufpr.br/letras/article/view/19407>. Acesso em: 17 abr. 2019.

NEUMANN, S.; HANSEN-SCHIRRA, S. Corpus methodology and design. *In*: HANSEN-SCHIRRA, S.; NEUMANN, S.; STEINER, E. **Cross-linguistic corpora for the study of translations**: Insights from the language pair English-German. Berlin: De Gruyter Mouton, 2012. p. 21-34. DOI: https://doi.org/10.1515/9783110260328.

NIVRE, J. Treebanks. *In*: KYTO, M., LUDELING, A.; MCENERY, T. (org.). **Corpus Linguistics**: An International Handbook. Berlin: De Gruyter Mouton, 2008. p. 225-241.

NOVODVORSKI, A.; FINATTO, M. J. B. Linguística de Corpus no Brasil: uma aventura mais do que adequada. **Letras & Letras**, Uberlândia, v. 30, n. 2, jul/dez. 2014. DOI: https://doi.org/10.14393/LL60-v30n2a2014-1. Disponível em: http://www.seer.ufu.br/index.php/letraseletras/article/viewFile/28516/15799. Acesso em: 10 abr. 2018.

OLIVEIRA, F. P. de. ToGatherUp. 2018.
Disponível em: www.togatherup.ileel.ufu.br. Acesso em: 1 mar. 2019.

O'REGAN, G. **A brief history of computing**. London: Springer Science & Business Media, 2008.

OTÁVIO, J. **Sublime Text IDE**: Introdução a melhor IDE para desenvolvimento. 2018. Disponível em: https://www.devmedia.com.br/sublime-text-ide-introducao-a-melhor-ide-para-desenvolvimento/34117 . Acesso em: 20 fev. 2019.

PAVEL, S.; NOLET, D. **Manual de terminologia**. Tradução Enilde Faulstich. Canadá: Departamento de Tradução, 2002.

PROJECT MANAGEMENT INSTITUTE (PMI). **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)**. 5. ed. Newtown Square: Project Management Institute, 2013.

PUSTEJOVSKY, J.; STUBBS, A. **Natural Language Annotation for Machine Learning**: A guide to corpus-building for applications. Sebastopol: O'Reilly Media, 2012.

RADFORD, A. *et al*. **Language models are unsupervised multitask learners**. Disponível em: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Acesso em: 15 abr. 2019.

RASO, T.; MELLO, H. (Org.). **Spoken corpora and linguistic studies**. 1ed. Amsterdam/Philadelphia: John Benjamins, 2014. DOI: https://doi.org/10.1075/scl.61.00int.

RENOUF, A. Corpus development 25 years on: from super-corpus to cybercorpus. *In*: FACCHINETTI, R. (org.) **Language and Computers**: Studies in Practical Linguistics, v. 62, n. 1, p. 27, 2007. DOI: https://doi.org/10.1163/9789401204347_004.

REPPEN, R. Building a corpus: What are the key considerations? *In*: O'KEEFFE, A.; MCCARTHY, M. J. (org.). **The Routledge handbook of corpus linguistics.** London**:** Routledge, 2010, p. 31-37.

RUBI, M. P. Os princípios da política de indexação na análise de assunto para catalogação: especificidade, exaustividade, revocação e precisão na perspectiva dos catalogadores e usuários. *In*: FUJITA, M. S. L. *et al*. **A indexação de livros:** a percepção de catalogadores e usuários de bibliotecas universitárias: um estudo de observação do contexto sociocognitivo com protocolos verbais. São Paulo: Cultura Acadêmica, 2009. p. 81-93.

RUMSEY, D. **Statistics essentials for dummies**. Hoboken: John Wiley & Sons, 2010.

RUNDELL, M.; KILGARRIFF, A. Automating the Creation of Dictionaries: Where Will It All End? *In*: MEUNIER, F. *et al*. (ed.). **A Taste for Corpora**: A tribute to Professor Sylviane Granger. Amsterdam: Benjamins, 2011. p. 257-281. DOI: https://doi.org/10.1075/scl.45.15run.

RYCHLÝ, P. NoSketch Engine. 2007. Brno: Masaryk University. Disponível em: https://nlp.fi.muni.cz/trac/noske. Acesso em: 5 dez. 2018.

SANTOS, A. **Contributions for building a Corpora-Flow system**. Supervision: José João Dias de Almeida e Anália Maria Garcia Lourenço. 2011. 100 f. (Master in Informatics Engineering) – Escola de Engenharia, Universidade do Minho, Braga, 2011. Disponível em: https://repositorium.sdum.uminho.pt/bitstream/1822/28122/1/eeum_di_dissertacao_pg15973.pdf. Acesso em: 17 abr. 2019.

SCHÄFER, R.; BILDHAUER, F. **Web Corpus Construction**. Toronto: University of Toronto, 2013. DOI: https://doi.org/10.2200/S00508ED1V01Y201305HLT022.

SHACKELFORD, R. *et al*. Computing curricula 2005: The overview report. **ACM SIGCHI Bulletin**. New York: ACM, v. 38, n. 1, 2006. p. 456-457. DOI: https://doi.org/10.1145/1121341.1121482. Disponível em: https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf. Acesso em: 11 abr. 2019.

SKINNER, J; BOND, W. Sublime Text 3. Version 3.1.1. [Computer Software]. Sydney: Sublime HQ Pty Ltd, 2013. Disponível em: https://www.sublimetext.com/. Acesso em: 9 abr. 2019.

SCHMID, H. Tokenizing and part-of-speech tagging. *In*: LÜDELING, A.; KYTÖ, M. (ed.). **Corpus linguistics:** An international handbook. Berlin: Mouton de Gruyter, 2008. p. 527-551.

SEDLAR, E. **Database-managed file system**. US Pat. US20050091287A1, 28 abr. 2005. 44 p.

SEMINO, E.; SHORT, M. **Corpus stylistics**: Speech, writing and thought presentation in a corpus of English writing. London: Routledge, 2004. DOI: https://doi.org/10.4324/9780203494073.

SIMOV, K. *et al*. CLaRK. Version 3.0. [Computer Software]. Tübingen: Tübingen Universitätsstadt, 2001. Disponível em: http://bultreebank.org/en/clark/. Acesso em: 5 dez. 2018.

SIMSKE, S. J. **Systems and methods for processing text-based electronic documents**. U.S. Patent n. 7,106, 905, 12 set. 2006.

SINCLAIR, J. McH. **Corpus, Concordance, Collocation**. Oxford: Oxford University Press, 1991.

SINCLAIR, J. McH. Preliminary recommendations on corpus typology. **EAGLES Document TCWG-CTYP/P.** 1996. Disponível em: http://www. ilc. pi. cnr. it/EAGLES/corpustyp/corpustyp. html. Acesso em: 30 maio 2018. Não paginado.

SINCLAIR, J. McH. Corpus and Text – Basic Principles. *In*: WYNNE, M. (ed.). **Developing linguistic corpora**: A Guide to Good Practice. Oxford: Oxbow Books, 2005. Disponível em: https://ota.ox.ac.uk/documents/creating/dlc/chapter1.htm. Acesso em: 2 abr. 2019.

SOUZA, G. P. **Linguagem de especialidade da ciência da informação**: estudo exploratório a partir dos periódicos brasileiros da área entre 2005 e 2009. Orientadora: Dra. Johanna Wilhelmina Smit. 2011. 104 f. Dissertação (Mestrado em Ciência da Informação) – Escola de Comunicações e Artes, Universidade de São Paulo, 2011. Disponível em: http://www.teses.usp.br/teses/disponiveis/27/27151/tde-06102015-103743/pt-br.php. Acesso em: 17 abr. 2019.

SWALES, J. M. **Genre Analysis**: English in Academic and Research Settings. Cidade: Cambridge University Press, 1990.

TAGNIN, S. E. O. Glossário de linguística de corpus. *In*: VIANA, V.; TAGNIN, S. E. O. (org.). **Cor-pora no ensino de línguas estrangeiras**. São Paulo: HUB Editorial, 2010. p. 349-353.

TAGNIN, S. E. O. Corpus-driven terminology in Brazil. *In*: POUPET, A. B.; XATARA, C. (org.). **Cahiers de lexicologie:** dynamique de la recherche en lexicologie, lexicographie et terminologie au Brésil. Paris: Classiques Garnier, v. 2012-2, n. 101. p. 169-182, 2012. Disponível em: https://classiques-garnier.com/cahiers-de-lexicologie-2012-2-n-101-dynamique-de-la-recherche-en-lexicologie-lexicographie-et-terminologie-au-bresil.html. Acesso em: 17 abr. 2019.

FFLCH/USP. **Projeto Corpus Multilíngue para Ensino e Tradução (CoMet)**. Disponível em: http://comet.fflch.usp.br. Acesso em: 17 abr. 2019.

TAGNIN, S. E. O. **Corpora na Tradução**, São Paulo: Hub Editorial, 2015.

TOGNINI-BONELLI, E. **Corpus linguistics at work**. Amsterdam: John Benjamins, 2001. 224 p. DOI: https://doi.org/10.1075/scl.6.

TURKEL, W. J.; CRYMBLE, A. Normalizing Textual Data with Python. 2012. Disponível em: https://programminghistorian.org/en/lessons/normalizing-data . Acesso em: 20 fev. 2019.

URDAN, T. C. **Statistics in plain English**. New York: Routledge, 2011.

VOORMANN, H.; GUT, U. Agile corpus creation. **Corpus Linguistics and Linguistic Theory**, Berlin, v. 4, n. 2, p. 235-251, 2008. DOI: https://doi.org/10.1515/CLLT.2008.010.

WEISSER, M. ICEweb (Version 2) [Computer Software]. China, Guangzhou: Guangdong University of Foreign Studies, 2008. Disponível em: http://martinweisser.org/ling_soft.html. Acesso em: 5 dez. 2018.

WIDDOWSON, H.G. **Linguistics**. Oxford: Oxford University Press, 1996.

W3SCHOOLS. XML Tree. 2019. Disponível em: https://www.w3schools.com/xml/xml_tree.asp. Acesso em: 8 nov. 2019.

ZANETTIN, F. **Translation-driven corpora**: Corpus resources for descriptive and applied translation studies. London: Routledge, 2014. DOI: https://doi.org/10.4324/9781315759661.

APÊNDICE A – Relatórios UltraLex

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: AntCorGen

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 20:22
Última modificação:	24/01/2019 21:31

Informações gerais sobre o produto

Nome da obra/programa:	AntCorGen
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://www.laurenceanthony.net/software/antcorgen/
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Computador
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	TXT
Subcategoria:	Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade)
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Fácil de instalar Gratuito Instalável em Windows Salva dados

	Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos Interface em língua estrangeira Layout final rígido Monolíngue Não importa dados Poucas funcionalidades
Observações:	O AntCorGen é uma ferramenta para coleta automática de corpora a partir dos dados disponíveis publicados na revista PLOS ONE (https://journals.plos.org/plosone/), uma revista eletrônica multidisciplinar, por meio de consultas à base de dados do periódico e download automático de arquivos em formato TXT. Na definição da consulta, o usuário deve informar a expressão que será usada como parâmetro da consulta pela ferramenta e as áreas da PLOS ONE que deverão ser consultadas. O *corpus* gerado é constituído de textos nomeados automaticamente de acordo com padrão definido pelo programa e não pelo usuário. Os arquivos do *corpus* são salvos em diretórios sem relação com as áreas selecionadas na consulta. O AntCorGen possui a capacidade de coletar dados automaticamente como os web crawlers, mas difere-se desse tipo de ferramenta devido ao escopo restrito de sua pesquisa.
Informações sobre *corpus*	
Ferramentas de *Corpus*:	Clusters Download de *corpus*
Ferramentas Auxiliares de *Corpus*:	BootCat/Extrator de *corpus*
Suporte para construção manual de *Corpus*:	Conversão de dados para formato TXT

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: AntFileSplitter

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 09:01
Última modificação:	25/01/2019 09:17

Informações gerais sobre o produto

Nome da obra/programa:	AntFileSplitter
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://www.laurenceanthony.net/software/antfilesplitter/
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Computador
Tipo de projeto:	Individual
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	TXT
Formatos de Importação:	TXT
Subcategoria:	Lex - recurso Term - recurso
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Fácil de instalar Gratuito Importa dados Instalável em Windows

	Salva dados Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos Interface em língua estrangeira Layout final rígido Poucas funcionalidades
Observações:	Conforme a documentação da ferramenta (cf. http://www.laurenceanthony.net/software/antfilesplitter/releases/AntFileSplitter100/help.pdf), o AntFileSplitter é um recurso para a divisão de arquivos de texto em subconjuntos de arquivos com a quantidade de tokens determinada pelo usuário. O AntFileSplitter gera como resultado, conjuntos de arquivos nomeados automaticamente (de acordo com padrão definido pelo sistema. Ex: results_001.txt, results_002.txt, n...). Cada um dos arquivos resultantes do processamento do AntFileSplitter contém a quantidade de tokens definida pelo usuário, no momento da configuração da divisão. O AntFileSplitter faz a remoção de etiquetagem dos textos.

Informações sobre *corpus*

Ferramentas de *Corpus*:	Download de *corpus* Upload de *corpus*

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: BootCat

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 09:31
Última modificação:	25/01/2019 09:50

Informações gerais sobre o produto

Nome da obra/programa:	BootCaT
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://bootcat.dipintra.it
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Computador
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	TXT
Subcategoria:	*Corpus* - ferramenta simples Lex - ferramenta simples Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Fácil de instalar Gratuito Instalável em Windows Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos

	Interface em língua estrangeira Layout final rígido Poucas funcionalidades
Informações sobre *corpus*	
Ferramentas de *Corpus*:	Download de *corpus*
Ferramentas Auxiliares de *Corpus*:	BootCat/Extrator de *corpus*
Suporte para construção manual de *Corpus*:	Conversão de dados para formato TXT

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: CLaRK

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 09:59
Ultima modificação:	25/01/2019 11:18

Informações gerais sobre o produto

Nome da obra/programa:	CLaRK
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://bultreebank.org/bg/clark/
Categoria:	Ambiente
Tipo de usuário:	Misto
Complexidade:	Alta
Ambiente de Trabalho:	Computador
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	XML
Formatos de Importação:	XML
Subcategoria:	*Corpus* - múltiplas ferramentas Lex - múltiplas ferramentas Term - múltiplas ferramentas
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Fácil de instalar Gratuito Importa dados

	Instalável em Windows Múltiplas funcionalidades Salva dados Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos Interface em língua estrangeira Layout final rígido
Observações:	No site de distribuição do CLaRK (cf. http://bultreebank.org/en/clark/), na descrição da ferramenta, encontramos a informação de que o CLaRK é um sistema para desenvolvimento de corpora, cujo principal objetivo é a minimizar a intervenção humana na criação de recursos linguísticos. Do nosso ponto de vista, a ferramenta não contribui para a redução da intervenção humana, conforme a afirmação mencionada, pelas seguintes razões: a) a ferramenta somente processa dados em formato XML, ou seja, o linguísta precisará converter os dados para esse padrão, o que não é uma tarefa simples, antes de qualquer processamento linguístico pela ferramenta; b) para a criação de linhas de concordância ou de listas de palavras, o linguísta precisará dominar a XPath, a linguagem de consulta utilizada pela ferramenta para a realização de pesquisas no *corpus*; c) de modo geral, todas funcionalidades da ferramenta demandam a compreensão de conjuntos de conceitos e detalhes técnicos que podem ser enigmáticos e atrapalhar o trabalho do linguísta, ao invés de facilitá-lo; d) O CLaRK não possui interface intuitiva e amigável para o usuário.

Informações sobre *corpus*

Ferramentas de *Corpus*:	Clusters Colocados Concordanciador Criação de listas a partir de corpora disponíveis Download de *corpus* Etiquetador Lista de palavras Upload de *corpus*
Ferramentas Auxiliares de *Corpus*:	Visualizador de arquivos

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: ICEweb

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	25/01/2019 11:41
Última modificação:	25/01/2019 11:51

Informações gerais sobre o produto

Nome da obra/programa:	ICEWeb
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://martinweisser.org/ling_soft.html
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Computador
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	HTML TXT XML
Subcategoria:	*Corpus* - ferramenta simples Lex - ferramenta simples Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Exporta dados Fácil de instalar Gratuito Instalável em Windows Salva dados Trabalha com grandes corpora

Desvantagens:	Fechado para novos campos Interface em língua estrangeira Layout final rígido Não importa dados Poucas funcionalidades Sem manual de instrução
Observações:	O ICEWeb é uma ferramenta para coleta automática de dados (web crawler) e construção de web corpora. A ferramenta foi criada com o objetivo de ampliar o International *Corpus* of English (ICE) (cf. https://www.ice-corpora.uzh.ch/en.html), um *corpus* para estudos comparativos da Língua inglesa produzido por esforço conjunto de pesquisados de diferentes países.

Informações sobre *corpus*

Ferramentas de *Corpus*:	Concordanciador Download de *corpus* N-gramas
Ferramentas Auxiliares de *Corpus*:	BootCat/Extrator de *corpus* Visualizador de arquivos
Suporte para construção manual de *Corpus*:	Conversão de dados para formato TXT

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: NoSketch Engine

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 11:59
Última modificação:	25/01/2019 12:42

Informações gerais sobre o produto

Nome da obra/programa:	NoSketch Engine
Baseado em *Corpus*:	Sim
Endereço na Internet:	https://nlp.fi.muni.cz/trac/noske
Categoria:	Ambiente
Tipo de usuário:	Misto
Complexidade:	Alta
Ambiente de Trabalho:	Internet
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Subcategoria:	Ambiente de visualização
Público Alvo:	Especialistas
Vantagens:	Acesso aberto Exporta dados Gratuito Salva dados Trabalha com grandes corpora
Desvantagens:	Difícil de instalar Fechado para novos campos Interface em língua estrangeira Layout final rígido Não exporta dados Não importa dados Não instalável em Windows

	Não salva dados Poucas funcionalidades Sem manual de instrução
Observações:	O NoSketch Engine é uma versão gratuita e limitada (cf. https://www.sketchengine.eu/nosketch-engine/) do Sketch Engine. A ferramenta não possui corpora pré-carregados e não oferece suporte para a criação de corpora (user corpora ou web crawling). De acordo com a documentação do NoSketch Engine (cf. https://www.sketchengine.eu/nosketch-engine/), a intalação (https://nlp.fi.muni.cz/trac/noske/wiki/Downloads), o gerenciamento e o carregamento de corpora na ferramenta dependem de conhecimentos técnicos e de ferramentas de terceiros.

Informações sobre *corpus*

Ferramentas de *Corpus*:	Concordanciador Estatísticas Lista de palavras

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: Sketch Engine

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 08:26
Última modificação:	24/01/2019 19:34

Informações gerais sobre o produto

Nome da obra/programa:	Sketch Engine
Baseado em *Corpus*:	Sim
Endereço na Internet:	https://www.sketchengine.co.uk/
Categoria:	Ambiente
Tipo de usuário:	Misto
Complexidade:	Média
Ambiente de Trabalho:	Internet
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Importação:	BZ2 DOC DOCX GZ HTM HTML ODT PDF TAR TEI TGZ TMX TXT VERT XLF XLIFF XLS

	XLSX XML ZIP
Subcategoria:	*Corpus* - múltiplas ferramentas Lex - ambiente de ferramentas Lex - múltiplas ferramentas Term - ambiente de ferramentas
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Importa dados Multilíngue Múltiplas funcionalidades Salva dados Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos Layout final rígido Pago
Observações:	O Sketch Engine permite que seus usuários construam seu próprio *corpus* por meio do upload dos dados do corpora na interface da ferramenta. Porém, para que o programa possa aplicar as funcionalidades que dependem de metadados, o usuário precisará inserí-los, manualmente, no "vertical file" e no "registry file" (cf. https://www.sketchengine.eu/documentation/local-installations/compiling-*corpus*/), os arquivos que guardam as informações o corpora (conteúdo e definições). A documentação do Sketch Engine (cf. https://www.sketchengine.eu/documentation/text-types-headers-and-subcorpora/) informa de modo explícito que a ferramenta não consegue criar "hierarchical headers", as estruturas da arquitetura da ferramenta que comportam informações como os metadados. A documentação informa, ainda, que o usuário deverá habilitar o "expert mode" para a edição do "vertical file" e do "registry file" no *Corpus* Architect, a interface da ferramenta que dá suporte para a construção e gerenciamento de corpora. O Sketch Engine possui uma interface de troca de dados entre aplicações por meio do JSON (JavaScript Object Notation, http://www.json.org/). As formas de utilização da API estão disponíveis na documentação da ferramenta (https://www.sketchengine.co.uk/documentation/json-api-documentation/). O Sketch Engine oferece o "*Corpus*

	info", uma interface que exibe informações e estatísticas do *corpus* (cf. https://www.sketchengine.eu/user-guide/user-manual/corpora/*corpus*-statistics-and-details/). No entanto, as informações não são exibidas de forma estratificada, de acordo, por exemplo, com os metatados do *corpus*. O Sketch Engine permite o download dos corpora (cf. https://www.sketchengine.eu/user-guide/user-manual/corpora/download-a-*corpus*/) desde que tenham sido criados pelo próprio usuário. O *corpus* pode ser baixado em formato TXT ou no formato "vertical file". O Sketch Engine não oferece suporte para a nomeação de arquivos com base em convenções definidas pelo usuário. O Sketch Engine não oferece suporte para a exportação do *corpus* com base em definições hierárquicas do usuário.
Informações sobre *corpus*	
Ferramentas de *Corpus*:	Clusters Colocados Compartilhamento de *corpus* Concordanciador Criação de listas a partir de corpora disponíveis Download de *corpus* Estatísticas Etiquetador Extrator de arquivos compactados Extrator de palavras-chave Lematização Lista de palavras N-gramas Padrões de Associação Palavras-chave Upload de *corpus* Word Sketch Word Sketch bilíngue Word Sketch Difference
Ferramentas Auxiliares de *Corpus*:	Alinhamento corpora paralelos Análise estatística Análise sincrônica/diacrônica BootCat/Extrator de *corpus* Comparação de corpora paralelos Conversor de formatos de textos Detector de *corpus* corrompido Lemma list Match list Modelo de configuração de *corpus* Padrões Stop list

	Tesaurus
Suporte para construção manual de *Corpus*:	Estatísticas da coleta de dados do *Corpus* Inclusão, exclusão e edição de dados do *Corpus*

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: Sub-*corpus* Creator

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	24/10/2017 23:24
Última modificação:	25/01/2019 23:32

Informações gerais sobre o produto

Nome da obra/programa:	Sub-*corpus* Creator
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://*corpus*.bfsu.edu.cn/tools
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Computador
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	TXT
Formatos de Importação:	TXT
Subcategoria:	*Corpus* - ferramenta simples Lex - ferramenta simples Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Exporta dados Fácil de instalar Gratuito Importa dados Instalável em Windows

	Salva dados Trabalha com grandes corpora
Desvantagens:	Fechado para novos campos Interface em língua estrangeira Layout final rígido Poucas funcionalidades Sem manual de instrução
Observações:	O Sub-*corpus* Creator é uma ferramenta que cria sub-corpora a partir de corpora carregados em sua interface por meio de filtros estabelecidos pelo usuário. Os arquivos dos sub-corpora são filtrados a partir de partes de seus nomes ou por segmentos do seu conteúdo. O Sub-*corpus* Creator trabalha somente com dados no formato TXT, possui uma interface muito simples e limitada e todo o material referente à ferramenta disponível na Internet está em Língua chinesa.
Informações sobre *corpus*	
Ferramentas de *Corpus*:	Download de *corpus* Upload de *corpus*
Ferramentas Auxiliares de *Corpus*:	Visualizador de arquivos

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: TextDirectory

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	25/10/2017 06:34
Última modificação:	26/01/2019 06:39

Informações gerais sobre o produto

Nome da obra/programa:	TextDirectory
Baseado em *Corpus*:	Sim
Endereço na Internet:	https://textdirectory.readthedocs.io
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Internet
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	TXT
Formatos de Importação:	TXT
Subcategoria:	*Corpus* - ferramenta simples Lex - ferramenta simples Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto Com manual de instrução Exporta dados Gratuito Importa dados

Desvantagens:	Difícil de instalar Fechado para novos campos Interface em língua estrangeira Layout final rígido Não instalável em Windows Não salva dados Poucas funcionalidades
Observações:	O TextDirectory é um agregador de arquivos de texto. Os textos são agregados de acordo com filtros definidos pelo usuário (por tamanho, conteúdo ou aleatoriamente). O uso da ferramenta se dá por meio de linhas de comandos (Command-Line Tool) em consoles (bash ou PowerShell) e exige conhecimentos técnicos avançados do usuário.

Informações sobre *corpus*

Ferramentas de *Corpus*:	Download de *corpus*

UltraLex (ultralex.ileel.ufu.br)
Relatório de avaliação: TextSTAT

Informações gerais

Desenvolvedor:	Fernando Oliveira
Criação:	25/10/2017 07:21
Última modificação:	26/01/2019 07:25

Informações gerais sobre o produto

Nome da obra/programa:	TextSTAT
Baseado em *Corpus*:	Sim
Endereço na Internet:	http://neon.niederlandistik.fu-berlin.de/de/textstat/
Categoria:	Recurso
Tipo de usuário:	Misto
Complexidade:	Baixa
Ambiente de Trabalho:	Comp. e Internet
Tipo de projeto:	Indiv./Colab.
Alvo principal:	Lexicografia/Terminografia
Quantidade de campos:	Fixa
Formatos de Exportação:	CSV DOC TXT XLS
Formatos de Importação:	DOC DOCX ODT TXT
Subcategoria:	*Corpus* - ferramenta simples Lex - ferramenta simples Term -ferramenta simples
Público Alvo:	Aprendizes (especialidade) Especialistas
Vantagens:	Acesso aberto

	Com manual de instrução Exporta dados Fácil de instalar Gratuito Instalável em Windows Salva dados Trabalha com grandes corpora
Desvantagens:	Difícil de instalar Fechado para novos campos Interface em língua estrangeira Layout final rígido
Observações:	O TextSTAT é um programa concordanciador simples. A ferramenta permite o carregamento de corpora já existentes em sua interface. A ferramenta possui função de coleta automática de textos da Internet a partir de SOMENTE um único endereço de site especificado pelo usuário. O TextSTAT possui uma versão instalável em Windows (Fácil de instalar) e uma versão para uso na Internet, cuja instalação depende de conhecimentos técnicos avançados (Difícil de instalar).
Informações sobre *corpus*	
Ferramentas de *Corpus*:	Concordanciador Criação de listas a partir de corpora disponíveis Extrator de palavras-chave Lista de palavras Palavras-chave Upload de *corpus*
Ferramentas Auxiliares de *Corpus*:	BootCat/Extrator de *corpus*
Suporte para construção manual de *Corpus*:	Conversão de dados para formato TXT

APÊNDICE B – Primeira proposta de Árvore de Domínio da Computação

Computing

1. Algorithms and Complexity
2. Artificial Intelligence
2.1. Cognitive Science
2.1.1. Machine Learning
2.1.1.1. Supervised Learning
2.1.1.2. Unsupervised Learning
2.1.1.3. Reinforcement Learning
2.1.2. Speech Recognition
2.1.3. Computer Vision
2.1.4. Natural Language Processing (NLP)
2.2. Common Sense Knowledge
2.3. Evolutionary Computation
2.4. Expert Systems
2.5. Search
2.6. Reasoning
2.7. Inferece
2.8. Knowledge Representation
2.9. Perception
2.10. Planning
2.11. Predictive Analytics
2.12. Robotics
2.13. Social Intelligence
3. Biomedical Computing
3.1. Computer Surgery
4. Computational Linguistics
4.1. Computational Discourse
4.2. Computational Morphology
4.3. Computational Phonology
4.4. Computational Semantics
4.5. Computational Syntax
5. Computational Neuroscience
5.1. Neural Encoding Models
5.2. Neural Coding
5.3. Neural Decoding
5.4. Neural Models
5.5. Computing in Carbon
5.6. Plasticity in the Brain & Learning
6. Computational Sociology
7. Computer Architecture
7.1. Computers
7.1.1. Desktop
7.1.1.1. Towers
7.1.1.2. All-in-One Computers
7.1.1.3. Minis
7.1.2. Game Consoles
7.1.3. Laptops
7.1.3.1. Traditional Laptops
7.1.3.2. 2 in 1s
7.1.4. Mainframes
7.1.5. Mainframes computers
7.1.6. Midrange computers
7.1.7. Mobile Devices
7.1.8. Quantum Computers
7.1.9. Servers
7.1.10. Supercomputers
7.1.11. Workstations
7.2. Hardware
7.2.1. Computer Cards
7.2.2. Memories
7.2.3. Peripherals
7.2.3.1. Cases
7.2.3.2. Energy Devices
7.2.3.3. Input and Output (I/O)
7.2.3.4. Storage devices
7.2.3.4.1. Optical storage Devices
7.2.3.4.2. Magnetic storage devices
7.2.3.4.3. Flash memory devices
7.2.3.4.4. Cloud storage
7.2.3.4.5. Digital Media
7.2.3.5. Processors
8. Computer Engineering
9. Computer Graphics
9.1. Computer Vision
9.2. Animation
9.3. Computational geometry
9.4. Geometric Modeling
9.5. Image processing
9.6. Scientific visualization
9.7. Visual Simulation
9.8. Image Synthesis
9.9. Virtual/Augmented Reality
9.10. Computational Visualization
10. Computer Networks
10.1. Network Devices
10.2. Network Types
10.3. Cloud Computing
10.3.1. Applications and Services
10.3.2. Cloud Clients
10.3.3. Platform and Storage Infrastructure
10.3.4. Computing Infrastructure
10.4. Intercloud
10.5. Internet, World Wide Web
10.6. Internet of Things (IoT)
10.6.1. Wearable Technologies
10.6.2. Wearable Devices
10.6.3. Wearable Sensors
11. Computer Security
11.1. Authentication and Authorisation
11.2. Biometrics
11.3. Cloud Computing Security
11.4. Cryptography
11.5. Ethical Hacking
11.6. Data security
11.7. Defensive Programming
11.8. Digital Forensics
11.9. Information Security
11.9.1. Social Engineering
11.10. Platform Security
11.11. Privacy
11.12. Security Engineering
11.13. Security Policy, Laws and Computer Crimes

11.14. Threats and Attacks
11.15. Web Security
12. Data Structures
13. Distributed Computing
14. Human-Computer Interaction
15. Information Science
15.1. Data
15.1.1. Data Management
15.1.2. Data Mining
15.1.3. Data Modeling
15.1.4. Multimedia Systems
15.1.5. Information retrieval
15.1.6. Knowledge management
15.1.7. Multimedia, hypermedia
15.1.8. Information Management
15.1.8.1. Information Models
15.1.9. Knowledge Management
15.1.10. Multimedia
15.1.11. Database
15.1.11.1. Relational Databases
15.1.11.2. Distributed Database
15.1.11.3. Object Database
15.1.12. Database Systems
15.1.13. Big Data
15.1.13.1. Data Analytics
15.1.13.2. Text Analytics
15.1.13.3. Web Analytics
15.1.13.4. Network Analytics
15.1.13.5. Mobile Analytics
15.1.14. Data Systems
15.1.14.1. Data Science
15.1.15. Data Warehouse
15.1.16. Visual Data Analysis
16. Information Systems
16.1. Information Technology
16.2. Management Information Systems
16.3. Health Information Systems
17. Parallel Computing
17.1. High-performance computing
18. Programming Languages
18.1. Program Semantics
18.2. Programming Paradigms
18.2.1. Object-oriented programming
18.2.2. Functional programming
18.2.3. Concurrent programming
18.3. Compilers
19. Software Engineering
19.1. Game Development
19.1.1. Computer animation
19.1.2. Game Architecture
19.1.3. Game Design
19.1.4. Game Engine Development
19.1.5. Game programming
19.2. Simulation
19.3. Software Architecture
19.4. Software Design
19.5. Software Development
19.6. Software Engineering
19.7. Software Platforms
19.8. Software Processes
19.9. Software Verification
19.10. Software Validation
19.11. Software Evolution
19.12. Risk Assessment
19.12.1. Game Platforms
19.12.2. Industrial Platforms
19.12.3. Mobile Platforms
19.12.4. Web Platforms
19.13. Software Quality Assurance
19.14. Software Reliability
19.15. Software Requirements
19.16. Software Testing
19.17. Software Types
19.17.1. Applications
19.17.1.1. Desktop Applications
19.17.1.2. Mobile Aplications
19.17.1.3. Web application
19.17.2. Device Drives
19.17.3. Games
19.18. Operating Systems
19.19. User Experience (UX)
19.19.1. Human-Computer Interface (HCI)
19.19.2. Information Architecture
19.19.3. Information Design
19.19.4. Interection Design
19.19.5. Usability Design
19.19.6. User Experience Design
19.19.7. User Interfaces (UI)
19.19.8. Visual Design
20. Theory of Computation
21. Net-Centric Computing
21.1. Network Communication
21.2. Network Security
21.3. Web Organization
21.4. Networked Applications
21.5. Network Management
21.6. Multimedia Technologies
21.7. Mobile Computing
22. Quantum Computing
23. Computer Training
23.1. Blogs
23.2. Communities
23.3. Courses
23.4. Chats
23.5. Documentation
23.6. FAQ
23.7. Foruns
23.8. Helpers
23.9. Manuals
23.10. Product Support
23.11. Tutorials
23.12. Wiki
24. Computer Certifications
25. Social Issues and Professional Practice
25.1. Intellectual Property
25.2. Privacy and Civil Liberties
25.3. Professional Communication
25.3.1. Reports
25.3.2. Diagrams
25.3.3. Technical Notes

25.4. Professional Ethics
25.5. Sustainability
26. Community Informatics
27. Computational Biology
28. Computational Chemistry
29. Computational Economics
30. Computational Finance
31. Computational Fluid Dynamics
32. Computational Mathematics
33. Computational Number Theory
34. Computational Physics
35. Computer Aided Engineering
36. Humanistic Informatics
37. Humanities Computing
38. Scientific Computing

APÊNDICE C – Segunda proposta de Árvore de Domínio da Computação

Computing

1. HARDWARE
1.1. Computer Architecture
1.1.1. Computers
1.1.1.1. Desktops
1.1.1.2. Game Consoles
1.1.1.3. Laptops
1.1.1.4. Mainframes computers
1.1.1.5. Midrange computers
1.1.1.6.Mobile Devices
1.1.1.7.Quantum Computers
1.1.1.8.Servers
1.1.1.9.Supercomputers
1.1.1.10.Workstations
1.1.2..Peripherals
1.1.2.1.Energy Devices
1.1.2.2.Input and Output (I/O)
1.1.2.3.Storage devices
1.1.3.Processors
1.1.4.Computer Cards
1.1.5.Memories
1.1.5.1.Caches
1.1.5.2.Physical Memory
1.1.5.3.Virtual Memory
1.2.Computer Engineering
1.3.Quantum Computing
1.4.Distributed Computing
1.5.Parallel Computing
2.SOFTWARE
2.1.Software Development
2.1.1.Game Development
2.1.2.Programming
2.1.3.Software Architecture
2.1.4.Software Development Techniques
2.1.5.Software Engineering
2.1.6.Software Platforms
2.1.7.User Experience (UX)
2.2.Data Management
2.2.1.Big Data
2.2.2.Data Analysis
2.2.3.Data Analytics
2.2.4.Data architecture
2.2.5.Data governance
2.2.6.Data Modeling
2.2.7.Data Quality
2.2.8.Data Security
2.2.9.Database Systems
2.2.10.Databases
2.2.11.Meta Data Management
2.3.Computer Graphics
2.3.1.Computational Geometry
2.3.2.Computer Animation
2.3.3.Image processing
2.3.4.Scientific Visualization
2.3.5.Virtual/Augmented Reality
2.3.6.Visual Simulation
2.3.7.Visualization
2.4.Software Types
2.4.1.Applications
2.4.2.Device Drives
2.4.3.Games
2.4.4.Operating Systems
2.5.Cybersecurity
2.5.1.Environmental Security
2.5.2.Ethical Hacking
2.5.3.Information Security
2.5.4.Network Security
2.5.5.Operations Security
2.5.6.Physical Security
2.5.7.Security Architecture
2.5.8.Security Governance
2.5.9.Security Systems
2.5.10.Threats Intelligence
2.6.Net-Centric Computing
2.6.1.Cloud Computing
2.6.2.Internet of Things (IoT)
2.6.3.Mobile Computing
2.6.4.Network Communication
2.6.5.Network Devices
2.6.6.Network Management
2.6.7.Network Types
2.6.8.Networked Applications
2.6.9.Web Organization
2.7.Artificial Intelligence
2.7.1.Computer Vision
2.7.2.Expert Systems
2.7.3.Learning Systems
2.7.4.Machine Learning
2.7.5.Natural Language Processing (NLP)
2.7.6.Perceptive Systems
2.7.7.Planning
2.7.8.Reasoning
2.7.9.Robotics
2.7.10.Speech Recognition
3.TRAINING/EDUCATION
3.1.Blogs
3.2.Career Development
3.3.Chats
3.4.Communities
3.5.Computer Certifications
3.6.Courses
3.7.Distance education
3.8.Documentation
3.9.e-Learning
3.10.FAQ
3.11.Foruns
3.12.Helpers
3.13.Manuals
3.14.Product Support
3.15.Professional Communication
3.16.Tutorials
3.17.Wiki
4.APPLIED COMPUTING

4.1.Biomedical Computing
4.2.Computational Biology
4.3.Computational Chemistry
4.4.Computational Economics
4.5.Computational Finance
4.6.Computational Fluid Dynamics
4.7.Computational Linguistics
4.8.Computational Mathematics
4.9.Computational Neuroscience
4.10.Computational Physics
4.11.Computational Sociology
4.12.Computer Aided Engineering
4.13.Humanistic Informatics
4.14.Scientific Computing
4.15.Human-computer interaction

ANEXO A – Computing Classification System (CSS)

Com a exclusão dos itens “General and reference” e “Proper nouns: People, technologies and companies” por não serem de áreas da Computação.

Computing Classification System, 2012 Revision
Association for Computing Machinery
30 March 2012

- Hardware
  - Printed circuit boards
    - Electromagnetic interference and compatibility
    - PCB design and layout
  - Communication hardware, interfaces and storage
    - Signal processing systems
      - Digital signal processing
      - Beamforming
      - Noise reduction
    - Sensors and actuators
    - Buses and high-speed links
    - Displays and imagers
    - External storage
    - Networking hardware
    - Printers
    - Sensor applications and deployments
    - Sensor devices and platforms
    - Sound-based input / output
    - Tactile and hand-based interfaces
      - Touch screens
      - Haptic devices
    - Scanners
    - Wireless devices
    - Wireless integrated network sensors
    - Electro-mechanical devices
  - Integrated circuits
    - 3D integrated circuits
    - Interconnect
      - Input / output circuits
      - Metallic interconnect
      - Photonic and optical interconnect
      - Radio frequency and wireless interconnect
    - Semiconductor memory
      - Dynamic memory
      - Static memory
      - Non-volatile memory
      - Read-only memory
    - Digital switches
      - Transistors
      - Logic families
    - Logic circuits
      - Arithmetic and datapath circuits
      - Asynchronous circuits
      - Combinational circuits
      - Design modules and hierarchy
      - Finite state machines
      - Sequential circuits
    - Reconfigurable logic and FPGAs
      - Hardware accelerators
      - High-speed input / output
      - Programmable logic elements
      - Programmable interconnect
      - Reconfigurable logic applications
  - Very large scale integration design
    - 3D integrated circuits
    - Analog and mixed-signal circuits
      - Data conversion
      - Clock generation and timing
      - Analog and mixed-signal circuit optimization
      - Radio frequency and wireless circuits
      - Wireline communication

- Analog and mixed-signal circuit synthesis
- Application-specific VLSI designs
  - Application specific integrated circuits
  - Application specific instruction set processors
  - Application specific processors
- Design reuse and communication-based design
  - Network on chip
  - System on a chip
  - Platform-based design
  - Hard and soft IP
- Design rules
- Economics of chip design and manufacturing
- Full-custom circuits
- VLSI design manufacturing considerations
- On-chip resource management
- On-chip sensors
- Standard cell libraries
- VLSI packaging
  - Die and wafer stacking
  - Input / output styles
  - Multichip modules
  - Package-level interconnect
- VLSI system specification and constraints

Power and energy
- Thermal issues
  - Temperature monitoring
  - Temperature simulation and estimation
  - Temperature control
  - Temperature optimization
- Energy generation and storage
  - Batteries
  - Fuel-based energy
  - Renewable energy
  - Reusable energy storage
- Energy distribution
  - Energy metering
  - Power conversion
  - Power networks
  - Smart grid
- Impact on the environment
- Power estimation and optimization
  - Switching devices power issues
  - Interconnect power issues
  - Circuits power issues
  - Chip-level power issues
  - Platform power issues
  - Enterprise level and data centers power issues

Electronic design automation
- High-level and register-transfer level synthesis
  - Datapath optimization
  - Hardware-software codesign
  - Resource binding and sharing
  - Operations scheduling
- Hardware description languages and compilation
- Logic synthesis
  - Combinational synthesis
  - Circuit optimization
  - Sequential synthesis
  - Technology-mapping
  - Transistor-level synthesis
- Modeling and parameter extraction
- Physical design (EDA)
  - Clock-network synthesis
  - Packaging
  - Partitioning and floorplanning
  - Placement
  - Physical synthesis
  - Power grid design
  - Wire routing
- Timing analysis
  - Electrical-level simulation
  - Model-order reduction
  - Compact delay models
  - Static timing analysis
  - Statistical timing analysis
  - Transition-based timing analysis

- Methodologies for EDA
  - Best practices for EDA
  - Design databases for EDA
  - Software tools for EDA

Hardware validation
- Functional verification
  - Model checking
  - Coverage metrics
  - Equivalence checking
  - Semi-formal verification
  - Simulation and emulation
  - Transaction-level verification
  - Theorem proving and SAT solving
  - Assertion checking
- Physical verification
  - Design rule checking
  - Layout-versus-schematics
  - Power and thermal analysis
  - Timing analysis and sign-off
- Post-manufacture validation and debug
  - Bug detection, localization and diagnosis
  - Bug fixing (hardware)
  - Design for debug

Hardware test
- Analog, mixed-signal and radio frequency test
- Board- and system-level test
- Defect-based test
- Design for testability
  - Built-in self-test
  - Online test and diagnostics
  - Test data compression
- Fault models and test metrics
- Memory test and repair
- Hardware reliability screening
- Test-pattern generation and fault simulation
- Testing with distributed and parallel systems

Robustness
- Fault tolerance
  - Error detection and error correction
  - Failure prediction
  - Failure recovery, maintenance and self-repair
  - Redundancy
  - Self-checking mechanisms
  - System-level fault tolerance
- Design for manufacturability
  - Process variations
  - Yield and cost modeling
  - Yield and cost optimization
- Hardware reliability
  - Aging of circuits and systems
  - Circuit hardening
  - Early-life failures and infant mortality
  - Process, voltage and temperature variations
  - Signal integrity and noise analysis
  - Transient errors and upsets
- Safety critical systems

Emerging technologies
- Analysis and design of emerging devices and systems
  - Emerging architectures
  - Emerging languages and compilers
  - Emerging simulation
  - Emerging tools and methodologies
- Biology-related information processing
  - Bio-embedded electronics
  - Neural systems
- Circuit substrates
  - III-V compounds
  - Carbon based electronics
  - Cellular neural networks
  - Flexible and printable circuits
  - Superconducting circuits
- Electromechanical systems
  - Microelectromechanical systems
  - Nanoelectromechanical systems
- Emerging interfaces
- Memory and dense storage

- Emerging optical and photonic technologies
- Reversible logic
- Plasmonics
- Quantum technologies
  - Single electron devices
  - Tunneling devices
  - Quantum computation
    - Quantum communication and cryptography
    - Quantum error correction and fault tolerance
  - Quantum dots and cellular automata
- Spintronics and magnetic technologies

Computer systems organization

- Architectures
  - Serial architectures
    - Reduced instruction set computing
    - Complex instruction set computing
    - Superscalar architectures
    - Pipeline computing
    - Stack machines
  - Parallel architectures
    - Very long instruction word
    - Interconnection architectures
    - Multiple instruction, multiple data
    - Cellular architectures
    - Multiple instruction, single data
    - Single instruction, multiple data
    - Systolic arrays
    - Multicore architectures
  - Distributed architectures
    - Cloud computing
    - Client-server architectures
    - n-tier architectures
    - Peer-to-peer architectures
    - Grid computing
  - Other architectures
    - Neural networks
    - Reconfigurable computing
    - Analog computers
    - Data flow architectures
    - Heterogeneous (hybrid) systems
    - Self-organizing autonomic computing
    - Optical computing
    - Quantum computing
    - Molecular computing
    - High-level language architectures
    - Special purpose systems
- Embedded and cyber-physical systems
  - Sensor networks
  - Robotics
    - Robotic components
    - Robotic control
    - Robotic autonomy
    - External interfaces for robotics
  - Sensors and actuators
  - System on a chip
  - Embedded systems
    - Firmware
    - Embedded hardware
    - Embedded software
- Real-time systems
  - Real-time operating systems
  - Real-time languages
  - Real-time system specification
  - Real-time system architecture
- Dependable and fault-tolerant systems and networks
  - Reliability
  - Availability
  - Maintainability and maintenance
  - Processors and memory architectures
  - Secondary storage organization
  - Redundancy
  - Fault-tolerant network topologies

- Networks
  - Network architectures
    - Network design principles
      - Layering
      - Naming and addressing
    - Programming interfaces
  - Network protocols
    - Network protocol design
    - Protocol correctness
      - Protocol testing and verification
      - Formal specifications
    - Link-layer protocols
    - Network layer protocols
      - Routing protocols
      - Signaling protocols
    - Transport protocols
    - Session protocols
    - Presentation protocols
    - Application layer protocols
      - Peer-to-peer protocols
    - OAM protocols
      - Time synchronization protocols
      - Network policy
    - Cross-layer protocols
    - Network File System (NFS) protocol
  - Network components
    - Intermediate nodes
      - Routers
      - Bridges and switches
    - Physical links
      - Repeaters
    - Middle boxes / network appliances
    - End nodes
      - Network adapters
      - Network servers
    - Wireless access points, base stations and infrastructure
      - Cognitive radios
    - Logical nodes
      - Network domains
  - Network algorithms
    - Data path algorithms
      - Packet classification
      - Deep packet inspection
      - Packet scheduling
    - Control path algorithms
      - Network resources allocation
      - Network control algorithms
      - Traffic engineering algorithms
      - Network design and planning algorithms
    - Network economics
  - Network performance evaluation
    - Network performance modeling
    - Network simulations
    - Network experimentation
    - Network performance analysis
    - Network measurement
  - Network properties
    - Network security
      - Security protocols
      - Web protocol security
      - Mobile and wireless security
      - Denial-of-service attacks
      - Firewalls
    - Network range
      - Short-range networks
      - Local area networks
      - Metropolitan area networks
      - Wide area networks
      - Very long-range networks
    - Network structure
      - Topology analysis and generation
      - Physical topologies
      - Logical / virtual topologies
      - Network topology types

- Point-to-point networks
- Bus networks
- Star networks
- Ring networks
  - Token ring networks
  - Fiber distributed data interface (FDDI)
- Mesh networks
  - Wireless mesh networks
- Hybrid networks

- Network dynamics
- Network reliability
  - Error detection and error correction
- Network mobility
- Network manageability
- Network privacy and anonymity

- Network services
  - Naming and addressing
  - Cloud computing
  - Location based services
  - Programmable networks
  - In-network processing
  - Network management
  - Network monitoring
- Network types
  - Network on chip
  - Home networks
  - Storage area networks
  - Data center networks
  - Wired access networks
  - Cyber-physical networks
    - Sensor networks
  - Mobile networks
  - Overlay and other logical network structures
    - Peer-to-peer networks
    - World Wide Web (network structure)
    - Social media networks
    - Online social networks
  - Wireless access networks
    - Wireless local area networks
    - Wireless personal area networks
  - Ad hoc networks
    - Mobile ad hoc networks
  - Public Internet
  - Packet-switching networks

Software and its engineering

- Software organization and properties
  - Contextual software domains
    - E-commerce infrastructure
    - Software infrastructure
      - Interpreters
      - Middleware
        - Message oriented middleware
        - Reflective middleware
        - Embedded middleware
      - Virtual machines
    - Operating systems
      - File systems management
      - Memory management
        - Virtual memory
        - Main memory
        - Allocation / deallocation strategies
        - Garbage collection
        - Distributed memory
        - Secondary storage
      - Process management
        - Scheduling
        - Deadlocks
        - Multithreading
        - Multiprocessing / multiprogramming / multitasking
        - Monitors
        - Mutual exclusion
        - Concurrency control
        - Power management

- Process synchronization
- Communications management
  - Buffering
  - Input / output
  - Message passing
- Virtual worlds software
  - Interactive games
  - Virtual worlds training simulations

Software system structures
- Embedded software
- Software architectures
  - n-tier architectures
  - Peer-to-peer architectures
  - Data flow architectures
  - Cooperating communicating processes
  - Layered systems
  - Publish-subscribe / event-based architectures
  - Electronic blackboards
  - Simulator / interpreter
  - Object oriented architectures
  - Tightly coupled architectures
  - Space-based architectures
  - 3-tier architectures
- Software system models
  - Petri nets
  - State systems
  - Entity relationship modeling
  - Model-driven software engineering
  - Feature interaction
  - Massively parallel systems
- Ultra-large-scale systems
- Distributed systems organizing principles
  - Cloud computing
  - Client-server architectures
  - Grid computing
  - Organizing principles for web applications
- Real-time systems software
- Abstraction, modeling and modularity

Software functional properties
- Correctness
  - Synchronization
  - Functionality
  - Real-time schedulability
  - Consistency
  - Completeness
  - Access protection
- Formal methods
  - Model checking
  - Software verification
  - Automated static analysis
  - Dynamic analysis

Extra-functional properties
- Interoperability
- Software performance
- Software reliability
- Software fault tolerance
  - Checkpoint / restart
- Software safety
- Software usability

Software notations and tools

General programming languages
- Language types
  - Parallel programming languages
  - Distributed programming languages
  - Imperative languages
  - Object oriented languages
  - Functional languages
  - Concurrent programming languages
  - Constraint and logic languages
  - Data flow languages
  - Extensible languages
  - Assembly languages
  - Multiparadigm languages
  - Very high level languages
- Language features
  - Abstract data types

- Polymorphism
- Inheritance
- Control structures
- Data types and structures
- Classes and objects
- Modules / packages
- Constraints
- Recursion
- Concurrent programming structures
- Procedures, functions and subroutines
- Patterns
- Coroutines
- Frameworks

- Formal language definitions
  - Syntax
  - Semantics
- Compilers
  - Interpreters
  - Incremental compilers
  - Retargetable compilers
  - Just-in-time compilers
  - Dynamic compilers
  - Translator writing systems and compiler generators
  - Source code generation
  - Runtime environments
  - Preprocessors
  - Parsers
- Context specific languages
  - Markup languages
    - Extensible Markup Language (XML)
    - Hypertext languages
  - Scripting languages
  - Domain specific languages
  - Specialized application languages
  - API languages
  - Graphical user interface languages
  - Window managers
  - Command and control languages
  - Macro languages
  - Programming by example
  - State based definitions
  - Visual languages
  - Interface definition languages
- System description languages
  - Design languages
  - Unified Modeling Language (UML)
  - Architecture description languages
  - System modeling languages
  - Orchestration languages
  - Integration frameworks
  - Specification languages
- Development frameworks and environments
  - Object oriented frameworks
  - Software as a service orchestration systems
  - Integrated and visual development environments
  - Application specific development environments
- Software configuration management and version control systems
- Software libraries and repositories
- Software maintenance tools

- Software creation and management
  - Designing software
    - Requirements analysis
    - Software design engineering
    - Software design tradeoffs
    - Software implementation planning
      - Software design techniques
  - Software development process management
    - Software development methods
      - Rapid application development
      - Agile software development
      - Capability Maturity Model
      - Waterfall model
      - Spiral model
      - V-model
      - Design patterns
    - Risk management

- Software development techniques
  - Software prototyping
  - Object oriented development
  - Flowcharts
  - Reusability
    - Software product lines
  - Error handling and recovery
- Software verification and validation
  - Software prototyping
  - Operational analysis
  - Software defect analysis
    - Software testing and debugging
  - Fault tree analysis
  - Process validation
    - Walkthroughs
    - Pair programming
    - Use cases
    - Acceptance testing
    - Traceability
  - Formal software verification
  - Empirical software validation
- Software post-development issues
  - Software reverse engineering
  - Documentation
  - Backup procedures
  - Software evolution
  - Software version control
  - Maintaining software
  - System administration
- Collaboration in software development
  - Open source model
  - Programming teams

Theory of computation

- Models of computation
  - Computability
    - Lambda calculus
    - Turing machines
    - Recursive functions
  - Probabilistic computation
  - Quantum computation theory
    - Quantum complexity theory
    - Quantum communication complexity
    - Quantum query complexity
    - Quantum information theory
  - Interactive computation
  - Streaming models
  - Concurrency
    - Parallel computing models
    - Distributed computing models
    - Process calculi
  - Timed and hybrid models
  - Abstract machines
- Formal languages and automata theory
  - Formalisms
    - Algebraic language theory
    - Rewrite systems
  - Automata over infinite objects
  - Grammars and context-free languages
  - Tree languages
  - Automata extensions
    - Transducers
    - Quantitative automata
  - Regular languages
- Computational complexity and cryptography
  - Complexity classes
  - Problems, reductions and completeness
  - Communication complexity
  - Circuit complexity
  - Oracles and decision trees
  - Algebraic complexity theory
  - Quantum complexity theory
  - Proof complexity
  - Interactive proof systems

- Complexity theory and logic
- Cryptographic primitives
- Cryptographic protocols

Logic

- Logic and verification
- Proof theory
- Modal and temporal logics
- Automated reasoning
- Constraint and logic programming
- Constructive mathematics
- Description logics
- Equational logic and rewriting
- Finite Model Theory
- Higher order logic
- Linear logic
- Programming logic
- Abstraction
- Verification by model checking
- Type theory
- Hoare logic
- Separation logic

Design and analysis of algorithms

- Graph algorithms analysis
  - Network flows
  - Sparsification and spanners
  - Shortest paths
  - Dynamic graph algorithms
- Approximation algorithms analysis
  - Scheduling algorithms
  - Packing and covering problems
  - Routing and network design problems
  - Facility location and clustering
  - Rounding techniques
  - Stochastic approximation
  - Numeric approximation algorithms
- Mathematical optimization
  - Discrete optimization
    - Network optimization
  - Continuous optimization
    - Linear programming
    - Semidefinite programming
    - Convex optimization
    - Quasiconvex programming and unimodality
    - Stochastic control and optimization
    - Quadratic programming
    - Nonconvex optimization
  - Mixed discrete-continuous optimization
    - Submodular optimization and polymatroids
    - Integer programming
- Data structures design and analysis
  - Data compression
  - Pattern matching
  - Sorting and searching
  - Predecessor queries
  - Cell probe models and lower bounds
- Online algorithms
  - Online learning algorithms
    - Scheduling algorithms
  - Caching and paging algorithms
  - K-server algorithms
  - Adversary models
- Parameterized complexity and exact algorithms
  - Fixed parameter tractability
  - W hierarchy
- Streaming, sublinear and near linear time algorithms
  - Bloom filters and hashing
  - Sketching and sampling
  - Lower bounds and information complexity
  - Random order and robust communication complexity
  - Nearest neighbor algorithms
- Parallel algorithms
  - MapReduce algorithms
  - Self-organization
  - Shared memory algorithms
  - Vector / streaming algorithms
  - Massively parallel algorithms

- Distributed algorithms
  - MapReduce algorithms
  - Self-organization
- Algorithm design techniques
  - Backtracking
  - Branch-and-bound
  - Divide and conquer
  - Dynamic programming
  - Preconditioning
- Concurrent algorithms

Randomness, geometry and discrete structures

- Pseudorandomness and derandomization
- Computational geometry
- Generating random combinatorial structures
- Random walks and Markov chains
- Expander graphs and randomness extractors
- Error-correcting codes
- Random projections and metric embeddings
- Random network models

Theory and algorithms for application domains

- Machine learning theory
  - Sample complexity and generalization bounds
  - Boolean function learning
  - Unsupervised learning and clustering
  - Kernel methods
    - Support vector machines
    - Gaussian processes
  - Boosting
  - Bayesian analysis
  - Inductive inference
  - Online learning theory
  - Multi-agent learning
  - Models of learning
  - Query learning
  - Structured prediction
  - Reinforcement learning
    - Sequential decision making
    - Inverse reinforcement learning
    - Apprenticeship learning
    - Multi-agent reinforcement learning
    - Adversarial learning
  - Active learning
  - Semi-supervised learning
  - Markov decision processes
  - Regret bounds
- Algorithmic game theory and mechanism design
  - Social networks
  - Algorithmic game theory
  - Algorithmic mechanism design
  - Solution concepts in game theory
  - Exact and approximate computation of equilibria
  - Quality of equilibria
  - Convergence and learning in games
  - Market equilibria
  - Computational pricing and auctions
  - Representations of games and their complexity
  - Network games
  - Network formation
  - Computational advertising theory
- Database theory
  - Data exchange
  - Data provenance
  - Data modeling
  - Database query languages (principles)
  - Database constraints theory
  - Database interoperability
  - Data structures and algorithms for data management
  - Database query processing and optimization (theory)
  - Data integration
  - Logic and databases
  - Theory of database privacy and security
  - Incomplete, inconsistent, and uncertain databases

Semantics and reasoning

- Program constructs
  - Control primitives
  - Functional constructs

- Object oriented constructs
- Program schemes
- Type structures
- Program semantics
  - Algebraic semantics
  - Denotational semantics
  - Operational semantics
  - Axiomatic semantics
  - Action semantics
  - Categorical semantics
- Program reasoning
  - Invariants
  - Program specifications
  - Pre- and post-conditions
  - Program verification
  - Program analysis
  - Assertions
  - Parsing
  - Abstraction

Mathematics of computing

- Discrete mathematics
  - Combinatorics
    - Combinatoric problems
    - Permutations and combinations
    - Combinatorial algorithms
    - Generating functions
    - Combinatorial optimization
    - Combinatorics on words
    - Enumeration
  - Graph theory
    - Trees
    - Hypergraphs
    - Random graphs
    - Graph coloring
    - Paths and connectivity problems
    - Graph enumeration
    - Matchings and factors
    - Graphs and surfaces
    - Network flows
    - Spectra of graphs
    - Extremal graph theory
    - Matroids and greedoids
    - Graph algorithms
    - Approximation algorithms
- Probability and statistics
  - Probabilistic representations
    - Bayesian networks
    - Markov networks
    - Factor graphs
    - Decision diagrams
    - Equational models
    - Causal networks
    - Stochastic differential equations
    - Nonparametric representations
      - Kernel density estimators
      - Spline models
      - Bayesian nonparametric models
  - Probabilistic inference problems
    - Maximum likelihood estimation
    - Bayesian computation
    - Computing most probable explanation
    - Hypothesis testing and confidence interval computation
    - Density estimation
      - Quantile regression
    - Max marginal computation
  - Probabilistic reasoning algorithms
    - Variable elimination
    - Loopy belief propagation
    - Variational methods
    - Expectation maximization
    - Markov-chain Monte Carlo methods
      - Gibbs sampling
      - Metropolis-Hastings algorithm
      - Simulated annealing

- Markov-chain Monte Carlo convergence measures
- Sequential Monte Carlo methods
- Kalman filters and hidden Markov models
- Resampling methods
  - Bootstrapping
  - Jackknifing
- Random number generation
- Probabilistic algorithms
- Statistical paradigms
- Queueing theory
- Contingency table analysis
- Regression analysis
  - Robust regression
- Time series analysis
- Survival analysis
- Renewal theory
- Dimensionality reduction
- Cluster analysis
- Statistical graphics
- Exploratory data analysis
- Stochastic processes
  - Markov processes
- Nonparametric statistics
- Distribution functions
- Multivariate statistics

Mathematical software
- Solvers
- Statistical software
- Mathematical software performance

Information theory
- Coding theory

Mathematical analysis
- Numerical analysis
  - Computation of transforms
  - Computations in finite fields
  - Computations on matrices
  - Computations on polynomials
    - Gröbner bases and other special bases
  - Number-theoretic computations
  - Interpolation
  - Numerical differentiation
  - Interval arithmetic
  - Arbitrary-precision arithmetic
  - Automatic differentiation
  - Mesh generation
  - Discretization
- Mathematical optimization
  - Discrete optimization
    - Network optimization
  - Continuous optimization
    - Linear programming
    - Semidefinite programming
    - Convex optimization
    - Quasiconvex programming and unimodality
    - Stochastic control and optimization
    - Quadratic programming
    - Nonconvex optimization
  - Mixed discrete-continuous optimization
    - Submodular optimization and polymatroids
    - Integer programming
- Differential equations
  - Ordinary differential equations
  - Partial differential equations
  - Differential algebraic equations
  - Differential variational inequalities
- Calculus
  - Lambda calculus
  - Differential calculus
  - Integral calculus
- Functional analysis
  - Approximation
- Integral equations
- Nonlinear equations
- Quadrature

Continuous mathematics
- Calculus

- Lambda calculus
- Differential calculus
- Integral calculus
- Topology
  - Point-set topology
  - Algebraic topology
  - Geometric topology
- Continuous functions

Information systems

- Data management systems
  - Database design and models
    - Relational database model
    - Entity relationship models
    - Graph-based database models
      - Hierarchical data models
      - Network data models
    - Physical data models
    - Data model extensions
      - Semi-structured data
      - Data streams
      - Data provenance
      - Incomplete data
      - Temporal data
      - Uncertainty
      - Inconsistent data
  - Data structures
    - Data access methods
      - Multidimensional range search
      - Data scans
      - Point lookups
      - Unidimensional range search
      - Proximity search
    - Data layout
      - Data compression
      - Data encryption
      - Record and block layout
  - Database management system engines
    - DBMS engine architectures
    - Database query processing
      - Query optimization
      - Query operators
      - Query planning
      - Join algorithms
    - Database transaction processing
      - Data locking
      - Transaction logging
      - Database recovery
    - Record and buffer management
    - Parallel and distributed DBMSs
      - Key-value stores
      - MapReduce-based systems
      - Relational parallel and distributed DBMSs
    - Triggers and rules
    - Database views
    - Integrity checking
    - Distributed database transactions
      - Distributed data locking
      - Deadlocks
      - Distributed database recovery
    - Main memory engines
    - Online analytical processing engines
    - Stream management
  - Query languages
    - Relational database query languages
      - Structured Query Language
    - XML query languages
      - XPath
      - XQuery
    - Query languages for non-relational engines
      - MapReduce languages
    - Call level interfaces
  - Database administration
    - Database utilities and tools

- Database performance evaluation
- Autonomous database administration
- Data dictionaries
- Information integration
  - Deduplication
  - Extraction, transformation and loading
  - Data exchange
  - Data cleaning
  - Wrappers (data mining)
  - Mediators and data integration
  - Entity resolution
  - Data warehouses
  - Federated databases
- Middleware for databases
  - Database web servers
  - Application servers
  - Object-relational mapping facilities
  - Data federation tools
  - Data replication tools
  - Distributed transaction monitors
  - Message queues
  - Service buses
  - Enterprise application integration tools
  - Middleware business process managers

Information storage systems

- Information storage technologies
  - Magnetic disks
  - Magnetic tapes
  - Optical / magneto-optical disks
  - Storage class memory
    - Flash memory
    - Phase change memory
  - Disk arrays
  - Tape libraries
- Record storage systems
  - Record storage alternatives
    - Heap (data structure)
    - Hashed file organization
    - Indexed file organization
    - Linked lists
  - Directory structures
    - B-trees
    - Vnodes
    - Inodes
    - Extent-based file structures
  - Block / page strategies
    - Slotted pages
    - Intrapage space management
    - Interpage free-space management
  - Record layout alternatives
    - Fixed length attributes
    - Variable length attributes
    - Null values in records
  - Relational storage
    - Horizontal partitioning
    - Vertical partitioning
    - Column based storage
    - Hybrid storage layouts
    - Compression strategies
- Storage replication
  - Mirroring
  - RAID
  - Point-in-time copies
  - Remote replication
  - Storage recovery strategies
- Storage architectures
  - Cloud based storage
  - Storage network architectures
    - Storage area networks
    - Direct attached storage
    - Network attached storage
  - Distributed storage
- Storage management
  - Hierarchical storage management
  - Storage virtualization
  - Information lifecycle management

- Version management
- Storage power management
- Thin provisioning

Information systems applications
- Enterprise information systems
  - Intranets
  - Extranets
  - Enterprise resource planning
  - Enterprise applications
  - Data centers
- Collaborative and social computing systems and tools
  - Blogs
  - Wikis
  - Reputation systems
  - Open source software
  - Social networking sites
  - Social tagging systems
  - Synchronous editors
  - Asynchronous editors
- Spatial-temporal systems
  - Location based services
  - Geographic information systems
  - Sensor networks
  - Data streaming
  - Global positioning systems
- Decision support systems
  - Data warehouses
  - Expert systems
  - Data analytics
  - Online analytical processing
- Mobile information processing systems
- Process control systems
- Multimedia information systems
  - Multimedia databases
  - Multimedia streaming
  - Multimedia content creation
  - Massively multiplayer online games
- Data mining
  - Data cleaning
  - Collaborative filtering
  - Association rules
  - Clustering
  - Nearest-neighbor search
  - Data stream mining
- Digital libraries and archives
- Computational advertising
- Computing platforms

World Wide Web
- Web searching and information discovery
  - Web search engines
    - Web crawling
    - Web indexing
    - Page and site ranking
    - Spam detection
  - Content ranking
  - Collaborative filtering
  - Social recommendation
  - Personalization
  - Social tagging
- Online advertising
  - Sponsored search advertising
  - Content match advertising
  - Display advertising
  - Social advertising
- Web mining
  - Site wrapping
  - Data extraction and integration
    - Deep web
    - Surfacing
    - Search results deduplication
  - Web log analysis
  - Traffic analysis
- Web applications
  - Internet communications tools
    - Email
    - Blogs

- Texting
- Chat
- Web conferencing
- Social networks
- Crowdsourcing
  - Answer ranking
  - Trust
  - Incentive schemes
  - Reputation systems
- Electronic commerce
  - Digital cash
  - E-commerce infrastructure
  - Electronic data interchange
  - Electronic funds transfer
  - Online shopping
  - Online banking
  - Secure online transactions
  - Online auctions
- Web interfaces
  - Wikis
  - Browsers
  - Mashups
- Web services
  - Simple Object Access Protocol (SOAP)
  - RESTful web services
  - Web Services Description Language (WSDL)
  - Universal Description Discovery and Integration (UDDI)
  - Service discovery and interfaces
- Web data description languages
  - Semantic web description languages
    - Resource Description Framework (RDF)
    - Web Ontology Language (OWL)
  - Markup languages
    - Extensible Markup Language (XML)
    - Hypertext languages

Information retrieval

- Document representation
  - Document structure
  - Document topic models
  - Content analysis and feature selection
  - Data encoding and canonicalization
  - Document collection models
  - Ontologies
  - Dictionaries
  - Thesauri
- Information retrieval query processing
  - Query representation
  - Query intent
  - Query log analysis
  - Query suggestion
  - Query reformulation
- Users and interactive retrieval
  - Personalization
  - Task models
  - Search interfaces
  - Collaborative search
- Retrieval models and ranking
  - Rank aggregation
  - Probabilistic retrieval models
  - Language models
  - Similarity measures
  - Learning to rank
  - Combination, fusion and federated search
  - Information retrieval diversity
  - Top-k retrieval in databases
  - Novelty in information retrieval
- Retrieval tasks and goals
  - Question answering
  - Document filtering
  - Recommender systems
  - Information extraction
  - Sentiment analysis
  - Expert search
  - Near-duplicate and plagiarism detection
  - Clustering and classification
  - Summarization

- Business intelligence
- Evaluation of retrieval results
  - Test collections
  - Relevance assessment
  - Retrieval effectiveness
  - Retrieval efficiency
  - Presentation of retrieval results
- Search engine architectures and scalability
  - Search engine indexing
  - Search index compression
  - Distributed retrieval
  - Peer-to-peer retrieval
  - Retrieval on mobile devices
  - Adversarial retrieval
  - Link and co-citation analysis
  - Searching with auxiliary databases
- Specialized information retrieval
  - Structure and multilingual text search
    - Structured text search
    - Mathematics retrieval
    - Chemical and biochemical retrieval
    - Multilingual and cross-lingual retrieval
  - Multimedia and multimodal retrieval
    - Image search
    - Video search
    - Speech / audio search
    - Music retrieval
  - Environment-specific retrieval
    - Enterprise search
    - Desktop search
    - Web and social media search

Security and privacy

- Cryptography
  - Key management
  - Public key (asymmetric) techniques
    - Digital signatures
    - Public key encryption
  - Symmetric cryptography and hash functions
    - Block and stream ciphers
    - Hash functions and message authentication codes
  - Cryptanalysis and other attacks
  - Information-theoretic techniques
  - Mathematical foundations of cryptography
- Formal methods and theory of security
  - Trust frameworks
  - Security requirements
  - Formal security models
  - Logic and verification
- Security services
  - Authentication
    - Biometrics
    - Graphical / visual passwords
    - Multi-factor authentication
  - Access control
  - Pseudonymity, anonymity and untraceability
  - Privacy-preserving protocols
  - Digital rights management
  - Authorization
- Intrusion/anomaly detection and malware mitigation
  - Malware and its mitigation
  - Intrusion detection systems
  - Social engineering attacks
    - Spoofing attacks
    - Phishing
- Security in hardware
  - Tamper-proof and tamper-resistant designs
  - Embedded systems security
  - Hardware security implementation
    - Hardware-based security protocols
  - Hardware attacks and countermeasures
    - Malicious design modifications
    - Side-channel analysis and countermeasures
  - Hardware reverse engineering
- Systems security

- Operating systems security
  - Mobile platform security
  - Trusted computing
  - Virtualization and security
- Browser security
- Distributed systems security
- Information flow control
- Denial-of-service attacks
- Firewalls
- Vulnerability management
  - Penetration testing
  - Vulnerability scanners
- File system security

Network security
- Security protocols
- Web protocol security
- Mobile and wireless security
- Denial-of-service attacks
- Firewalls

Database and storage security
- Data anonymization and sanitization
- Management and querying of encrypted data
- Information accountability and usage control
- Database activity monitoring

Software and application security
- Software security engineering
- Web application security
- Social network security and privacy
- Domain-specific security and privacy architectures
- Software reverse engineering

Human and societal aspects of security and privacy
- Economics of security and privacy
- Social aspects of security and privacy
- Privacy protections
- Usability in security and privacy

Human-centered computing

Human computer interaction (HCI)
- HCI design and evaluation methods
  - User models
  - User studies
  - Usability testing
  - Heuristic evaluations
  - Walkthrough evaluations
  - Laboratory experiments
  - Field studies
- Interaction paradigms
  - Hypertext / hypermedia
  - Mixed / augmented reality
  - Command line interfaces
  - Graphical user interfaces
  - Virtual reality
  - Web-based interaction
  - Natural language interfaces
  - Collaborative interaction
- Interaction devices
  - Graphics input devices
  - Displays and imagers
  - Sound-based input / output
  - Keyboards
  - Pointing devices
  - Touch screens
  - Haptic devices
- HCI theory, concepts and models
- Interaction techniques
  - Auditory feedback
  - Text input
  - Pointing
  - Gestural input
- Interactive systems and tools
  - User interface management systems
  - User interface programming
  - User interface toolkits
- Empirical studies in HCI

- Interaction design
  - Interaction design process and methods
    - User interface design
    - User centered design
    - Activity centered design
    - Scenario-based design
    - Participatory design
    - Contextual design
    - Interface design prototyping
  - Interaction design theory, concepts and paradigms
  - Empirical studies in interaction design
  - Systems and tools for interaction design
    - Wireframes
- Collaborative and social computing
  - Collaborative and social computing theory, concepts and paradigms
    - Social content sharing
    - Collaborative content creation
    - Collaborative filtering
    - Social recommendation
    - Social networks
    - Social tagging
    - Computer supported cooperative work
    - Social engineering (social sciences)
    - Social navigation
    - Social media
  - Collaborative and social computing design and evaluation methods
    - Social network analysis
    - Ethnographic studies
  - Collaborative and social computing systems and tools
    - Blogs
    - Wikis
    - Reputation systems
    - Open source software
    - Social networking sites
    - Social tagging systems
    - Synchronous editors
    - Asynchronous editors
  - Empirical studies in collaborative and social computing
  - Collaborative and social computing devices
- Ubiquitous and mobile computing
  - Ubiquitous and mobile computing theory, concepts and paradigms
    - Ubiquitous computing
    - Mobile computing
    - Ambient intelligence
  - Ubiquitous and mobile computing systems and tools
  - Ubiquitous and mobile devices
    - Smartphones
    - Interactive whiteboards
    - Mobile phones
    - Mobile devices
    - Portable media players
    - Personal digital assistants
    - Handheld game consoles
    - E-book readers
    - Tablet computers
  - Ubiquitous and mobile computing design and evaluation methods
  - Empirical studies in ubiquitous and mobile computing
- Visualization
  - Visualization techniques
    - Treemaps
    - Hyperbolic trees
    - Heat maps
    - Graph drawings
    - Dendrograms
    - Cladograms
  - Visualization application domains
    - Scientific visualization
    - Visual analytics
    - Geographic visualization
    - Information visualization
  - Visualization systems and tools
    - Visualization toolkits
  - Visualization theory, concepts and paradigms
  - Empirical studies in visualization
  - Visualization design and evaluation methods
- Accessibility

- Accessibility theory, concepts and paradigms
- Empirical studies in accessibility
- Accessibility design and evaluation methods
- Accessibility technologies
- Accessibility systems and tools

Computing methodologies

- Symbolic and algebraic manipulation
  - Symbolic and algebraic algorithms
    - Combinatorial algorithms
    - Algebraic algorithms
    - Nonalgebraic algorithms
    - Symbolic calculus algorithms
    - Exact arithmetic algorithms
    - Hybrid symbolic-numeric methods
    - Discrete calculus algorithms
    - Number theory algorithms
    - Equation and inequality solving algorithms
    - Linear algebra algorithms
    - Theorem proving algorithms
    - Boolean algebra algorithms
    - Optimization algorithms
  - Computer algebra systems
    - Special-purpose algebraic systems
  - Representation of mathematical objects
    - Representation of exact numbers
    - Representation of mathematical functions
    - Representation of Boolean functions
    - Representation of polynomials
- Parallel computing methodologies
  - Parallel algorithms
    - MapReduce algorithms
    - Self-organization
    - Shared memory algorithms
    - Vector / streaming algorithms
    - Massively parallel algorithms
  - Parallel programming languages
- Artificial intelligence
  - Natural language processing
    - Information extraction
    - Machine translation
    - Discourse, dialogue and pragmatics
    - Natural language generation
    - Speech recognition
    - Lexical semantics
    - Phonology / morphology
    - Language resources
  - Knowledge representation and reasoning
    - Description logics
    - Semantic networks
    - Nonmonotonic, default reasoning and belief revision
    - Probabilistic reasoning
    - Vagueness and fuzzy logic
    - Causal reasoning and diagnostics
    - Temporal reasoning
    - Cognitive robotics
    - Ontology engineering
    - Logic programming and answer set programming
    - Spatial and physical reasoning
    - Reasoning about belief and knowledge
  - Planning and scheduling
    - Planning for deterministic actions
    - Planning under uncertainty
    - Multi-agent planning
    - Planning with abstraction and generalization
    - Robotic planning
  - Search methodologies
    - Heuristic function construction
    - Discrete space search
    - Continuous space search
    - Randomized search
    - Game tree search
    - Abstraction and micro-operators
    - Search with partial observations
  - Control methods
    - Robotic planning

- Computational control theory
- Motion path planning
- Philosophical/theoretical foundations of artificial intelligence
  - Cognitive science
  - Theory of mind
- Distributed artificial intelligence
  - Multi-agent systems
  - Intelligent agents
  - Mobile agents
  - Cooperation and coordination
- Computer vision
  - Computer vision tasks
    - Biometrics
    - Scene understanding
    - Activity recognition and understanding
    - Video summarization
    - Visual content-based indexing and retrieval
    - Visual inspection
    - Vision for robotics
    - Scene anomaly detection
  - Image and video acquisition
    - Camera calibration
    - Epipolar geometry
    - Computational photography
    - Hyperspectral imaging
    - Motion capture
    - 3D imaging
    - Active vision
  - Computer vision representations
    - Image representations
    - Shape representations
    - Appearance and texture representations
    - Hierarchical representations
  - Computer vision problems
    - Interest point and salient region detections
    - Image segmentation
    - Video segmentation
    - Shape inference
    - Object detection
    - Object recognition
    - Object identification
    - Tracking
    - Reconstruction
    - Matching

Machine learning

- Learning paradigms
  - Supervised learning
    - Ranking
    - Learning to rank
    - Supervised learning by classification
    - Supervised learning by regression
    - Structured outputs
    - Cost-sensitive learning
  - Unsupervised learning
    - Cluster analysis
    - Anomaly detection
    - Mixture modeling
    - Topic modeling
    - Source separation
    - Motif discovery
    - Dimensionality reduction and manifold learning
  - Reinforcement learning
    - Sequential decision making
    - Inverse reinforcement learning
    - Apprenticeship learning
    - Multi-agent reinforcement learning
    - Adversarial learning
  - Multi-task learning
    - Transfer learning
    - Lifelong machine learning
    - Learning under covariate shift
- Learning settings
  - Batch learning
  - Online learning settings
  - Learning from demonstrations
  - Learning from critiques

- Learning from implicit feedback
- Active learning settings
- Semi-supervised learning settings
- Machine learning approaches
  - Classification and regression trees
  - Kernel methods
    - Support vector machines
    - Gaussian processes
  - Neural networks
  - Logical and relational learning
    - Inductive logic learning
    - Statistical relational learning
  - Learning in probabilistic graphical models
    - Maximum likelihood modeling
    - Maximum entropy modeling
    - Maximum a posteriori modeling
    - Mixture models
    - Latent variable models
    - Bayesian network models
  - Learning linear models
    - Perceptron algorithm
  - Factorization methods
    - Non-negative matrix factorization
    - Factor analysis
    - Principal component analysis
    - Canonical correlation analysis
    - Latent Dirichlet allocation
  - Rule learning
  - Instance-based learning
  - Markov decision processes
  - Partially-observable Markov decision processes
  - Stochastic games
  - Learning latent representations
    - Deep belief networks
- Machine learning algorithms
  - Dynamic programming for Markov decision processes
    - Value iteration
    - Q-learning
    - Policy iteration
    - Temporal difference learning
    - Approximate dynamic programming methods
  - Ensemble methods
    - Boosting
    - Bagging
  - Spectral methods
  - Feature selection
  - Regularization
- Cross-validation

Modeling and simulation

- Model development and analysis
  - Modeling methodologies
  - Model verification and validation
  - Uncertainty quantification
- Simulation theory
  - Systems theory
  - Network science
- Simulation types and techniques
  - Uncertainty quantification
  - Quantum mechanic simulation
  - Molecular simulation
  - Rare-event simulation
  - Discrete-event simulation
  - Agent / discrete models
  - Distributed simulation
  - Continuous simulation
  - Continuous models
  - Real-time simulation
  - Interactive simulation
  - Multiscale systems
  - Massively parallel and high-performance simulations
  - Data assimilation
  - Scientific visualization
  - Visual analytics
  - Simulation by animation
- Simulation support systems
  - Simulation environments

- Simulation languages
- Simulation tools
- Simulation evaluation

Computer graphics
- Animation
  - Motion capture
  - Procedural animation
  - Physical simulation
  - Motion processing
  - Collision detection
- Rendering
  - Rasterization
  - Ray tracing
  - Non-photorealistic rendering
  - Reflectance modeling
  - Visibility
- Image manipulation
  - Computational photography
  - Image processing
  - Texturing
  - Image-based rendering
  - Antialiasing
- Graphics systems and interfaces
  - Graphics processors
  - Graphics input devices
  - Mixed / augmented reality
  - Perception
  - Graphics file formats
  - Virtual reality
- Image compression
- Shape modeling
  - Mesh models
  - Mesh geometry models
  - Parametric curve and surface models
  - Point-based models
  - Volumetric models
  - Shape analysis

Distributed computing methodologies
- Distributed algorithms
  - MapReduce algorithms
  - Self-organization
- Distributed programming languages

Concurrent computing methodologies
- Concurrent programming languages
- Concurrent algorithms

Applied computing

Electronic commerce
- Digital cash
- E-commerce infrastructure
- Electronic data interchange
- Electronic funds transfer
- Online shopping
- Online banking
- Secure online transactions
- Online auctions

Enterprise computing
- Enterprise information systems
  - Intranets
  - Extranets
  - Enterprise resource planning
  - Enterprise applications
  - Data centers
- Business process management
  - Business process modeling
  - Business process management systems
  - Business process monitoring
  - Cross-organizational business processes
  - Business intelligence
- Enterprise architectures
  - Enterprise architecture management
  - Enterprise architecture frameworks
  - Enterprise architecture modeling
- Service-oriented architectures

- Event-driven architectures
- Business rules
- Enterprise modeling
- Enterprise ontologies, taxonomies and vocabularies
- Enterprise data management
- Reference models
- Business-IT alignment
- IT architectures
- IT governance
- Enterprise computing infrastructures
- Enterprise interoperability
  - Enterprise application integration
  - Information integration and interoperability

Physical sciences and engineering

- Aerospace
  - Avionics
- Archaeology
- Astronomy
- Chemistry
- Earth and atmospheric sciences
  - Environmental sciences
- Engineering
  - Computer-aided design
- Physics
- Mathematics and statistics
- Electronics
  - Avionics
- Telecommunications
  - Internet telephony

Life and medical sciences

- Computational biology
  - Molecular sequence analysis
  - Recognition of genes and regulatory elements
  - Molecular evolution
  - Computational transcriptomics
  - Biological networks
  - Sequencing and genotyping technologies
  - Imaging
  - Computational proteomics
  - Molecular structural biology
  - Computational genomics
- Genomics
  - Computational genomics
- Systems biology
- Consumer health
- Health care information systems
- Health informatics
- Bioinformatics
- Metabolomics / metabonomics
- Genetics
  - Population genetics
  - Proteomics
    - Computational proteomics
  - Transcriptomics

Law, social and behavioral sciences

- Anthropology
  - Ethnography
- Law
- Psychology
- Economics
- Sociology

Computer forensics

- Surveillance mechanisms
- Investigation techniques
- Evidence collection, storage and analysis
- Network forensics
- System forensics
- Data recovery

Arts and humanities

- Fine arts
- Performing arts
- Architecture (buildings)
  - Computer-aided design
- Language translation
- Media arts
- Sound and music computing

- Computers in other domains
  - Digital libraries and archives
  - Publishing
  - Military
    - Cyberwarfare
  - Cartography
  - Agriculture
  - Computing in government
    - Voting / election technologies
    - E-government
  - Personal computers and PC applications
    - Word processors
    - Spreadsheets
    - Computer games
    - Microcomputers
- Operations research
  - Consumer products
  - Industry and manufacturing
    - Supply chain management
    - Command and control
  - Computer-aided manufacturing
  - Decision analysis
  - Transportation
  - Forecasting
  - Marketing
- Education
  - Digital libraries and archives
  - Computer-assisted instruction
  - Interactive learning environments
  - Collaborative learning
  - Learning management systems
  - Distance learning
  - E-learning
  - Computer-managed instruction
- Document management and text processing
  - Document searching
  - Document management
    - Text editing
    - Version control
    - Document metadata
  - Document capture
    - Document analysis
    - Document scanning
    - Graphics recognition and interpretation
    - Optical character recognition
    - Online handwriting recognition
  - Document preparation
    - Markup languages
      - Extensible Markup Language (XML)
      - Hypertext languages
    - Annotation
    - Format and notation
    - Multi / mixed media creation
    - Image composition
    - Hypertext / hypermedia creation
    - Document scripting languages

- Social and professional topics
  - Professional topics
    - Computing industry
      - Industry statistics
      - Computer manufacturing
      - Sustainability
    - Management of computing and information systems
      - Project and people management
        - Project management techniques
        - Project staffing
        - Systems planning
        - Systems analysis and design
        - Systems development
        - Computer and information systems training
      - Implementation management
        - Hardware selection
        - Computing equipment management
        - Pricing and resource allocation

- Software management
  - Software maintenance
  - Software selection and adaptation
- System management
  - Centralization / decentralization
  - Technology audits
  - Quality assurance
- Network operations
- File systems management
- Information system economics

- History of computing
  - Historical people
  - History of hardware
  - History of software
  - History of programming languages
  - History of computing theory

- Computing education
  - Computational thinking
  - Accreditation
  - Model curricula
  - Computing education programs
    - Information systems education
    - Computer science education
      - CS1
    - Computer engineering education
    - Information technology education
    - Information science education
    - Computational science and engineering education
    - Software engineering education
  - Informal education
  - Computing literacy
  - Student assessment
  - K-12 education
  - Adult education

- Computing and business
  - Employment issues
  - Automation
  - Computer supported cooperative work
  - Economic impact
  - Offshoring
  - Reengineering
  - Socio-technical systems

- Computing profession
  - Codes of ethics
  - Employment issues
  - Funding
  - Computing occupations
  - Computing organizations
  - Testing, certification and licensing
  - Assistive technologies

- Computing / technology policy
  - Intellectual property
    - Digital rights management
    - Copyrights
    - Software reverse engineering
    - Patents
    - Trademarks
    - Internet governance / domain names
    - Licensing
    - Treaties
    - Database protection laws
    - Secondary liability
    - Soft intellectual property
    - Hardware reverse engineering
  - Privacy policies
  - Censorship
    - Pornography
    - Hate speech
    - Political speech
    - Technology and censorship
    - Censoring filters
  - Surveillance
    - Governmental surveillance
    - Corporate surveillance
  - Commerce policy
    - Taxation

- Transborder data flow
- Antitrust and competition
- Governmental regulations
- Online auctions policy
- Consumer products policy
- Network access control
  - Censoring filters
  - Broadband access
  - Net neutrality
  - Network access restrictions
    - Age-based restrictions
    - Acceptable use policy restrictions
  - Universal access
- Computer crime
  - Social engineering attacks
    - Spoofing attacks
    - Phishing
  - Identity theft
  - Financial crime
  - Malware / spyware crime
- Government technology policy
  - Governmental regulations
  - Import / export controls
- Medical information policy
  - Medical records
  - Personal health records
  - Genetic information
  - Patient privacy
  - Health information exchanges
  - Medical technologies
    - Remote medicine

User characteristics

- Race and ethnicity
- Religious orientation
- Gender
  - Men
  - Women
- Sexual orientation
- People with disabilities
- Geographic characteristics
- Cultural characteristics
- Age
  - Children
  - Seniors
  - Adolescents

ANEXO B – Exemplo de texto produzido pelo modelo GPT-2

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	*The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.* *Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.* *Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.* *Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.* *Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.* *While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."* *Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.* *While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."* *However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.*

Disponível em: https://openai.com/blog/better-language-models/. Acesso em 15 abr. 2019.