SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 17/04/2007

Assinatura: _____

Avaliação automática da qualidade de escrita de resumos científicos em inglês

Luiz Carlos Genoves Junior

Orientadora: *Profa. Dra. Sandra Maria Aluísio*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. .

USP – São Carlos
Abril/2007

Avaliação automática da qualidade de
escrita de resumos científicos em
inglês

Luiz Carlos Genoves Jr.

Agradecimentos

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil.

Sumário

Lista de Figuras	p. ix
Lista de Tabelas	p. xiii
Resumo	p. xvii
Abstract	p. xix
1 Introdução	p. 1
2 Ferramentas de suporte à escrita de textos científicos baseadas na estrutura esquemática dos textos	p. 7
2.1 Estruturação de resumos acadêmicos	p. 7
2.2 SciPo – <i>Scientific Portuguese</i>	p. 11
2.3 SciPo-Farmácia	p. 15
2.4 Mover	p. 17
2.5 Considerações finais	p. 18
3 Conceitos de Aprendizado de Máquina, recursos e ferramentas de PLN	p. 21
3.1 Indução de classificadores	p. 21
3.1.1 Conceitos de aprendizado de máquina	p. 22
3.1.2 Algoritmos	p. 23
3.1.3 Métricas para a avaliação de classificadores	p. 26
3.1.3.1 Métodos para estimativa do erro real	p. 26

3.1.3.2	Estatísticas de erro nas classes	p. 27
3.1.3.3	Estatísticas de erro geral do classificador	p. 28
3.1.4	WEKA	p. 29
3.1.4.1	Indutores do WEKA	p. 33
3.2	Recursos lingüísticos para PLN: cópús e listas de frequência	p. 35
3.3	Ferramentas de PLN	p. 37
3.3.1	Sentenciador	p. 38
3.3.2	Tokenizador	p. 38
3.3.3	Etiquetador Morfossintático	p. 39
3.3.4	<i>Chunker</i>	p. 40
3.3.5	Ferramentas lingüísticas do pacote <i>OpenNLP</i>	p. 41
3.3.6	Conversores de formatos	p. 41
3.4	Considerações finais	p. 43
4	Métodos para detecção automática da estrutura esquemática de textos científicos	p. 45
4.1	<i>Argumentative Zoning</i>	p. 45
4.1.1	Avaliação Intrínseca	p. 46
4.2	<i>Argumentative Zoning for Portuguese (AZPort)</i>	p. 47
4.2.1	Avaliação Intrínseca	p. 48
4.3	O analisador automático do <i>Mover</i>	p. 49
4.3.1	Avaliação Intrínseca	p. 50
5	Qualidade de Escrita	p. 53
5.1	Histórico	p. 56
5.2	Técnicas utilizadas na avaliação automática da qualidade de escrita	p. 58
5.2.1	Técnicas de PLN	p. 58

5.2.2	Técnicas de Recuperação de Informação	p. 58
5.2.3	Técnicas Categorização Textual	p. 59
5.2.4	<i>Vector Space Model</i>	p. 59
5.2.5	Análise Semântica Latente	p. 60
5.2.6	Redes Complexas	p. 61
5.3	Descrição dos sistemas	p. 62
5.3.1	<i>E-rater</i>	p. 62
5.3.2	BETSY	p. 64
5.3.3	Sistema de Larkey (<i>Larkey's System</i>)	p. 65
5.3.3.1	Outros sistemas	p. 66
5.3.4	Categorização dos sistemas de avaliação da qualidade de escrita . .	p. 68
5.4	Uma rubrica específica do gênero científico para avaliação automática da qualidade de escrita	p. 69
5.5	Considerações finais	p. 73
6	Detecção automática da estrutura esquemática	p. 75
6.1	Pré-processamento do Córpus	p. 76
6.2	Esquema de anotação para resumos	p. 77
6.3	Atributos	p. 78
6.3.1	Contextuais	p. 79
6.3.2	Sintaxe Verbal	p. 81
6.3.3	Padrões Textuais	p. 82
6.3.3.1	Expressões-padrão de <i>abstracts</i>	p. 83
6.3.3.2	Formato das expressões	p. 84
6.3.4	Comprimento da sentença	p. 85
6.4	Treinamento e avaliação do <i>AZEA</i>	p. 85
6.4.1	Córpus	p. 86

6.4.1.1	Córpus de treinamento	p. 86
6.4.1.2	Córpus de teste	p. 87
6.4.2	Avaliação	p. 88
6.4.3	<i>Purpose Splitter</i>	p. 91
6.4.4	Comparação com outros sistemas	p. 94
6.4.4.1	Comparação com o mesmo córpus de treinamento	p. 94
6.5	<i>AZSections</i> : Estendendo a detecção para outras seções do texto científico	p. 96
6.5.1	Córpus	p. 97
6.5.2	Atributos	p. 98
6.5.3	Treinamento	p. 98
6.6	Considerações finais	p. 99

7 Implementando as dimensões de uma rubrica baseada no gênero científico p. 101

7.1	Implementações das dimensões relacionadas com organização e balanceamento	p. 101
7.2	Implementação de um detector automático de erros de uso de artigos	p. 103
7.2.1	Trabalhos relacionados sobre a detecção automática de erros de uso de artigo em inglês	p. 104
7.3	Detecção automática de erros de uso de artigos em resumos científico em inglês	p. 105
7.3.1	Córpus	p. 106
7.3.2	Atributos	p. 107
7.3.2.1	Contexto local	p. 107
7.3.2.2	Tipos de atributo	p. 108
7.3.3	Treinamento e avaliação	p. 111
7.3.3.1	Experimentos com os classificadores binários <i>HasArticle</i> e <i>DetArticle</i>	p. 113

7.4	Um classificador híbrido com dimensões da rubrica e técnicas de avaliação automática de qualidade de escrita: experimentos iniciais	p. 116
7.4.1	Atributos	p. 117
7.4.2	Extração dos atributos	p. 117
7.4.3	Treinamento e avaliação	p. 119
7.5	Considerações finais	p. 123
8	Conclusões	p. 125
	Referências	p. 129
	Apêndice A – Telas do <i>Criterion</i> (BURSTEIN et al., 2001, 2003)	p. 135
	Apêndice B – Erros gramaticais levantados na formulação da dimensão D5 da rubrica de resumos científicos (JR. et al.,)	p. 137
	Anexo A – Etiquetas morfossintáticas utilizadas no corpus <i>Penn Treebank</i> (MARCUS et al., 1993)	p. 141
	Anexo B – Rubrica usada na avaliação do GMAT (<i>AWA Scoring Guide</i>)	p. 143

Lista de Figuras

1	Modelo de um resumo típico segundo Weissberg & Buker (1990).	p. 8
2	Modelo de um resumo reduzido (WEISSBERG; BUKER, 1990).	p. 9
3	Componentes esquemáticas e suas respectivas estratégias retóricas (FELTRIM et al., 2002).	p. 9
4	Componentes esquemáticas e estratégias retóricas utilizadas no SciPo-Farmácia	p. 10
5	Modelo CARS modificado	p. 11
6	Visão geral do processo de estruturação do texto. Figura retirada de (FELTRIM, 2004). Na figura observamos o processo de crítica, que é cíclico, e composto dos itens (c) e (d)	p. 12
7	Crítica da estrutura. Um desvio grave foi detectado (em vermelho, falta da componente Resultados) e é sugerido que se insira as componentes Contexto e Lacuna.	p. 13
8	Recuperação de exemplos similares, em relação à estrutura esquemática. Cada uma das listas representa um grupo de textos similares, e a estrutura construída é mostrada abaixo do título “Estratégias escolhidas”.	p. 14
9	Arquitetura do ambiente SciPo. Figura retirada de (FELTRIM, 2004). . . .	p. 15
10	Exemplo de construção da estrutura esquemática para um resumo. As componentes esquemáticas e suas estratégias retóricas estão à esquerda. A janela no canto inferior direito é a resposta de uma requisição de ajuda do usuário sobre a definição da componente Conclusão.	p. 16
11	Texto classificado pelo <i>Mover</i>	p. 18
12	Exemplo de árvore de decisão. O nó A1 é o nó raiz, e os nós retangulares são terminais.	p. 24
13	Exemplo de conjunto de regras para o RIPPER.	p. 25

14	Exemplo de SVM linear.	p. 26
15	Matriz de Confusão	p. 27
16	Trecho de arquivo ARFF, retirado de http://www.cs.uu.nl/docs/vakken/dm/iris.arff	p. 30
17	Ambiente <i>Explorer</i> do WEKA sendo utilizado para ao treinamento de classificadores	p. 31
18	Entrada e saída do etiquetador morfossintático MXPOST. Separados por espaços estão os pares <i>token</i> -etiqueta, sendo que as etiquetas morfossintáticas estão em negrito. Trecho retirado do jornal <i>NY Times</i> (http://www.nytimes.com/2007/03/22/world/europe/22russia.html)	p. 40
19	Uma sentença separada em <i>chunks</i> . Os <i>chunks</i> são delimitados por '[' e ']', e suas funções sintáticas, que são apontadas no início do <i>chunk</i> . ADVP é um sintagma adverbial e PRT é uma partícula, sendo que o conjunto de etiquetas é descrito em (MARCUS et al., 1993).	p. 41
20	<i>Script</i> com aplicação do sentenciador, do tokenizador e do <i>chunker</i> do pacote <i>OpenNLP</i> para a extração dos <i>chunks</i> de um texto qualquer em inglês. . .	p. 42
21	Esquema de anotação usado no <i>Mover</i>	p. 50
22	Matriz de confusão do <i>Mover</i> com seu esquema original de anotação (ANTHONY; LASHKIA, 2003). Os valores entre parênteses indicam a precisão da classe	p. 51
23	Trecho da rubrica utilizada no GMAT para a avaliação de textos de alunos. São mostradas os requisitos para se obter a nota máxima (6) ou mínima (0).	p. 54
24	Evolução dos sistemas no tempo (MARÍN, 2004).	p. 57
25	Arquitetura do <i>E-rater</i>	p. 63
26	Os 11 atributos superficiais (<i>Text-complexity features</i>) utilizados no sistema de Larkey (LARKEY, 1998)	p. 66
27	Estrutura do processo de detecção automática da estrutura esquemática . .	p. 76
28	Parte da estrutura XML do resumo “01” do córpus do SciPo-Farmácia . . .	p. 87
29	Críticas das dimensões D1 e D2 fornecidas pelo <i>AZEA-Web</i>	p. 102

30	Contexto considerado na extração dos atributos	p. 108
31	35 palavras mais freqüentes utilizadas como valores do tipo de atributo <i>Palavra</i> . O token “ <i>t</i> ” representa um <i>not</i> contraído (como em <i>don’t</i>)	p. 109
32	Exemplo de extração dos valores dos atributos para o detector automático de erros de uso de artigo em inglês. Para simplificar, tratamos o trecho em questão como se fosse o texto sendo processado, e por isso o valor <i>new</i> do atributo <i>Discurso</i>	p. 111
33	Pré-processamento e extração dos valores dos atributos	p. 119
34	Árvore de decisão gerada pelo classificador J48, com a estrutura esquemática anotado manualmente.	p. 120
35	Resumo do desempenho de um aluno	p. 135
36	<i>Feedback</i> de organização	p. 136
37	Tabela de categorização de erros da rubrica	p. 138
38	Distribuição dos erros da dimensão D5 nos resumos dos alunos. Os 6 erros mais comuns estão destacados	p. 139

Lista de Tabelas

1	Como interpretar os valores da medida <i>Kappa</i> (LANDIS; KOCH, 1977) . . .	p. 29
2	Esquema de anotação do AZ (TEUFEL; MOENS, 2002).	p. 46
3	Atributos utilizados pelo AZ (TEUFEL; MOENS, 2002)	p. 47
4	Avaliação geral do AZ (TEUFEL; MOENS, 2002)	p. 48
5	Classes possíveis para cada sentença no <i>AZPort</i>	p. 48
6	Resumo dos atributos utilizados pelo <i>AZPort</i> na classificação de sentenças de resumos	p. 49
7	Avaliação do <i>AZPort</i>	p. 49
8	Sistemas divididos por: conteúdo ou estilo e <i>Rating Simulation</i> ou <i>Master Analysis</i> (VALENTI et al., 2003).	p. 69
9	Indicação das técnicas utilizadas pelos sistemas, resultados e onde foram aplicados (MARÍN, 2004). A sigla corr significa correlação; conc , concordância; acc , taxa de acerto. Quando os autores apresentam diversos valores, o menor foi o considerado	p. 69
10	Dimensões da rubrica. O <i>Kappa</i> relatado foi obtido entre anotadores <i>humanos</i> . A dimensão D5 foi reformulada	p. 73
11	Categorias usadas no <i>AZEA</i>	p. 78
12	Conjunto de atributos usados no <i>AZEA</i>	p. 79
13	Exemplo de extração de atributos contextuais <i>Localização</i> e <i>Histórico</i> . . .	p. 80
14	Estatísticas lexicais do corpus de treinamento, entre parênteses, temos o desvio padrão	p. 86
15	Distribuição das categorias no corpus de treinamento	p. 86
16	Distribuição das categorias no corpus <i>Atípicos</i>	p. 88

17	Distribuição das categorias no <i>córpus Alunos</i>	p. 88
18	Variando os algoritmos de AM supervisionado	p. 89
19	Aplicando o <i>AZEA</i> sobre os <i>córpus</i> de teste	p. 89
20	Contribuição dos atributos. Valores do Kappa	p. 90
21	Métricas estatísticas de avaliação de cada classe do <i>AZEA</i> usando SMO sobre <i>córpus</i> de treino	p. 90
22	Matriz de confusão para o <i>AZEA</i>	p. 91
23	Matriz de confusão para a classificação dos <i>córpus</i> de teste	p. 92
24	Ocorrência das categorias em relação à primeira sentença da categoria <i>Propósito</i>	p. 92
25	Estatísticas dos classificadores supondo que a extração do atributo <i>PurposePosition</i> fosse ótima	p. 93
26	Classificadores binários de identificação de sentenças da categoria <i>Propósito</i>	p. 93
27	Comparação dos sistemas baseados em <i>Zonas Argumentativas</i> . Valores relatados pelos autores	p. 94
28	Conjuntos de treino e teste usados na comparação com o <i>Mover</i>	p. 95
29	Comparação entre sistemas <i>AZEA</i> , <i>Mover</i> e <i>baseline</i>	p. 95
30	Matriz de confusão para o <i>Mover</i>	p. 96
31	Estatísticas detalhadas por categoria do <i>Mover</i> e do <i>AZEA</i>	p. 96
32	Estatísticas dos <i>córpus</i> utilizados no treinamento e avaliação do <i>AZSections</i>	p. 97
33	Categorias esquemáticas de cada um dos <i>córpus</i> do SciPo-Farmácia usados no <i>AZSections</i>	p. 97
34	Atributos utilizados no <i>AZSections</i>	p. 98
35	Precisão e <i>Kappa</i> para os <i>córpus</i> do SciPo-Farmácia	p. 99
36	Valores do atributo <i>Countability</i> no trabalho de (HAN et al., 2006).	p. 105
37	Estatísticas de cada texto do <i>córpus</i>	p. 106
38	Os 39 atributos, separados por grupos, utilizados na classificação	p. 110

39	Valores da taxa de acerto e <i>Kappa</i> sobre as três possíveis classes de artigos	p. 112
40	Precisão (P), Cobertura (C) e medida-F (F) para os classificadores sobre as três classes	p. 112
41	Classificadores treinados com o indutor J48 sobre cópús com três classes, separados por área (Farmácia e Física). São reportados os valores de acerto e o valor de <i>Kappa</i> entre parênteses	p. 112
42	Taxa de acerto e <i>Kappa</i> para os classificadores binários HA e DA	p. 114
43	Precisão, cobertura e medida-F para os classificadores HA (J48) e DA (NB)	p. 114
44	Contribuição dos conjuntos de atributos usados pelos classificadores binários	p. 114
45	Taxa de acerto e <i>Kappa</i> para os classificadores HA (J48) e DA (NB) quando aplicados ao cópús de resumos de alunos.	p. 115
46	Precisão, cobertura e medida-F para os classificadores HA (J48) e DA (NB) sobre os resumos de estudantes	p. 115
47	Cópús adaptados para os testes de detecção de resumos de boa qualidade	p. 116
48	Atributos utilizados pelos classificadores binários da qualidade de resumos	p. 118
49	Valores das métricas para com a estrutura esquemática real utilizada na extração dos atributos.	p. 120
50	Contribuição dos atributos. Valores do <i>Kappa</i> .	p. 121
51	Valores das métricas para o experimento com estrutura esquemática obtida pelo AZEA	p. 122
52	Resultados dos classificadores treinados com os cópús <i>azea</i> e <i>atípicos</i> e aplicados ao cópús de <i>corrigidos</i> (bons) e <i>estudantes</i> (ruins)	p. 122
53	Valores das métricas para o treinamento com todos os cópús, e os atributos obtidos da estrutura esquemática obtida pelo AZEA.	p. 123

Resumo

Problemas com a escrita podem afetar o desempenho de profissionais de maneira marcante, principalmente no caso de cientistas e acadêmicos que precisam escrever com proficiência e desembaraço não somente na língua materna, mas principalmente em inglês. Durante os últimos anos, ferramentas de suporte à escrita, algumas com enfoque em textos científicos, como o AMADEUS e o SciPo foram desenvolvidas e têm auxiliado pesquisadores na divulgação de suas pesquisas. Entretanto, a criação dessas ferramentas é baseada em *cópus*, sendo muito custosa, pois implica em selecionar textos bem escritos, além de segmentá-los de acordo com sua estrutura esquemática. Nesse mestrado estudamos, avaliamos e implementamos métodos de detecção automática da estrutura esquemática e de avaliação automática da qualidade de escrita de resumos científicos em inglês. Investigamos o uso de tais métodos para possibilitar o desenvolvimento de dois tipos de ferramentas: de detecção de bons resumos e de crítica. Nossa abordagem é baseada em *cópus* e em aprendizado de máquina supervisionado. Desenvolvemos um detector automático da estrutura esquemática, que chamamos de *AZEA*, com taxa de acerto de 80,4% e *Kappa* de 0,73, superiores ao estado da arte (acerto de 73%, *Kappa* de 0,65). Experimentamos várias combinações de algoritmos, atributos e diferentes seções de um artigo científicos. Utilizamos o *AZEA* na implementação de duas dimensões de uma rubrica para o gênero científico, composta de 7 dimensões, e construímos e disponibilizamos uma ferramenta de crítica da estrutura de um resumo. Um detector de erros de uso de artigo também foi desenvolvido, com precisão é de 83,7% (*Kappa* de 0,63) para a tarefa de decidir entre omitir ou não um artigo, com enfoque no *feedback* ao usuário e como parte da implementação da dimensão de erros gramaticais da rubrica. Na tarefa de detectar bons resumos, utilizamos métodos usados com sucesso na avaliação automática da qualidade de escrita de redações com as implementações da rubrica e realizamos experimentos iniciais, ainda com resultados fracos, próximos à *baseline*. Embora não tenhamos construído um bom avaliador automático da qualidade de escrita, acreditamos que este trabalho indica direções para atingir esta meta, e forneça algumas das ferramentas necessárias.

Abstract

Poor writing may have serious implications for a professional's career. This is even more serious in the case of scientists and academics whose job requires fluency and proficiency in their mother tongue as well as in English. This is why a number of writing tools have been developed in order to assist researchers to promote their work. Here, we are particularly interested in tools, such as AMADEUS and SciPo, which focus on scientific writing. AMADEUS and SciPo are corpus-based tools and hence they rely on corpus compilation which is by no means an easy task. In addition to the difficult task of selecting well-written texts, it also requires segmenting these texts according to their schematic structure. The present dissertation aims to investigate, evaluate and implement some methods to automatically detect the schematic structure of English abstracts and to automatically evaluate their quality. These methods have been examined with a view to enabling the development of two types of tools, namely: detection of well-written abstracts and a critique tool. For automatically detecting schematic structures, we have developed a tool, named AZEA, which adopts a corpus-based, supervised machine learning approach. AZEA reaches 80.4% accuracy and Kappa of 0.73, which is above the highest rates reported in the literature so far (73% accuracy and Kappa of 0.65). We have tested a number of different combinations of algorithms, features and different paper sections. AZEA has been used to implement two out of seven dimensions of a rubric for analyzing scientific papers. A critique tool for evaluating the structure of abstracts has also been developed and made available. In addition, our work also includes the development of a classifier for identifying errors related to English article usage. This classifier reaches 83.7% accuracy (Kappa of 0.63) in the task of deciding whether or not a given English noun phrase requires an article. If implemented in the dimension of grammatical errors of the above mentioned rubric, it can be used to give users feedback on their errors. As regards the task of detecting well-written abstracts, we have resorted to methods which have been successfully adopted to evaluate quality of essays and some preliminary tests have been carried out. However, our results are not yet satisfactory since they are not much above the baseline. Despite this drawback, we believe this study proves relevant since in addition to offering some of the necessary tools, it provides some fundamental guidelines towards the automatic evaluation of the quality of texts.

1 *Introdução*

Escrever não é fácil. Entretanto, é uma tarefa rotineira. Desde os primeiros anos de escola, somos requisitados a dissertar sobre temas variados, e independente da opinião que temos sobre o assunto¹, sempre somos avaliados pela qualidade do texto que escrevemos.

A avaliação da *qualidade de escrita* é uma tarefa árdua e subjetiva. No entanto, a habilidade de produzir *bons* textos é mensurada a todo momento, e influencia a vida de milhões de pessoas em todo o mundo. No Brasil, praticamente todos os vestibulares das faculdades brasileiras aplicam uma prova em que o aluno deve escrever uma redação, cuja qualidade é medida e usada, com outras notas, na classificação e eleição dos alunos que serão beneficiados com uma vaga. Empregos públicos são, quase sempre, obtidos por meio de concursos, os quais têm provas de avaliação da qualidade de escrita. Além disso, nesses casos de avaliação em larga escala, esta é feita por vários juízes, aumentando a subjetividade dessa tarefa.

Decisões tão importantes não podem ser tomadas sem embasamento nem ficarem à mercê da subjetividade da avaliação. A forma mais comum de conduzir este tipo de avaliação é definir um guia ou manual, no qual conste indicações e instruções de como proceder no julgamento da qualidade de um texto. Além disso, este julgamento deve ser validado, e a maneira mais comum é feita pela concordância entre os avaliadores, isto é, se diferentes avaliadores designam a mesma nota a um texto.

No meio acadêmico não é diferente. A comunicação entre as comunidades de pesquisa é feita pela publicação de artigos científicos. Estes artigos científicos são submetidos, aceitos e rejeitados a todo momento. Por isso, problemas com a escrita podem afetar o desempenho de pesquisadores e acadêmicos que precisam escrever com proficiência e desembaraço não apenas na língua materna, mas também em inglês, atualmente a *lingua franca* da ciência. Além do correto uso da língua, é muito importante conhecer as peculiaridades do gênero do texto que pretendemos escrever, para que este nosso texto atenda às expectativas dos

¹Embora aspectos éticos sejam também avaliados nas dissertações de vestibulares.

leitores desse gênero. Compartilhamos o conceito de gênero definido por Swales (1990):

“Um gênero compreende uma classe de eventos comunicativos, cujos membros compartilham o mesmo conjunto de propósitos comunicativos. Esses propósitos são reconhecidos pelos membros seniores da comunidade discursiva (leitores e escritores) e isso, simplesmente isso fundamenta um gênero. As amostras de um gênero exibem vários padrões de similaridade em termos de estrutura, estilo, conteúdo e público-alvo.”

Dentre as seções de um trabalho científico, certamente o resumo é uma das mais importantes, dado que muitos leitores se limitam a ler apenas o título e o resumo (BARRASS, 1979 apud FELTRIM, 2004)). Ainda, segundo Feltrim (2004):

“O resumo deve ser redigido com muito cuidado, de forma a ser completo, interessante e informativo, dispensando a consulta ao restante do texto para que o leitor tenha a idéia do que trata o trabalho e, ao mesmo tempo, e estimulando o interesse pela leitura do texto completo.”

Uma grande variedade de livros sobre escrita científica e ferramentas, como corretores ortográficos, gramaticais e estilísticos; ferramentas estatísticas, tais como contadores de palavras e sentenças; e corretores de erros mecânicos, fornecem diretrizes para escrever um texto e ajudam na pós-edição, mas normalmente falham ao tentar ajudar o autor a produzir o primeiro rascunho do texto.

Existem, entretanto, alguns trabalhos que detalham a estrutura do texto científico e as convenções lingüísticas como Swales (1990) e Weissberg & Buker (1990), que propõem modelos que descrevem a estrutura de resumos científicos. Estes modelos mostram a estrutura esperada do resumo, denominada **estrutura esquemática**. Tais modelos expõem os tipos de informação necessários e opcionais, e enfatizam que essas informações obedecem à uma ordem convencional de apresentação, esperada e observada em resumos científicos.

Baseados nesses estudos, pesquisadores do Núcleo Interinstitucional de Lingüística Computacional² (NILC) têm desenvolvido ferramentas computacionais para auxiliar a estruturação de um resumo e de outras seções de um texto científico, desde 1991. Exemplos de ferramentas são a ferramenta de suporte do AMADEUS (ALUÍSIO; O.N, 1995; ALUÍSIO; JR., 1996; ALUÍSIO; GANTENBEIN, 1997), o SciPo³ (FELTRIM, 2004) e o SciPo-Farmácia⁴ (ALUÍSIO

²<http://www.nilc.icmc.usp.br/nilc/index.html>.

³<http://www.nilc.icmc.usp.br/~scipo/>

⁴<http://www.nilc.icmc.usp.br/scipo-farmacia/>

et al., 2005; SCHUSTER et al., 2005). Essas ferramentas fornecem exemplos de estruturas de resumos científicos e ajudam o usuário a produzir o primeiro rascunho. Além disso, permitem que o usuário navegue na base de resumos, analise as sentenças separadas por funções retóricas dentro na estrutura esquemática e consulte marcadores discursivos usados para diversas funções. O SciPo ainda tem uma funcionalidade, baseada na ferramenta de crítica do AMADEUS, de crítica da estrutura esquemática, que questiona estruturas esquemáticas não convencionais construídas pelos usuários. A ferramenta de crítica também sugere a inserção de outras **componentes esquemáticas**, para o enriquecimento da estrutura. Componentes esquemáticas⁵ são sentenças que cumprem determinadas funções retóricas, como apresentar o propósito ou indicar uma lacuna, dentro da estrutura esquemática.

Estas ferramentas têm se demonstrado úteis, e têm sido usadas em cursos de escrita científica, em disciplinas de pós-graduação da USP, ministrados na Faculdade de Ciências Farmacêuticas em São Paulo e no Instituto de Física de São Carlos (IFSC). Estes cursos utilizam a ferramenta SciPo-Farmácia para ajudar os alunos na estruturação de seus textos científicos⁶ (o SciPo-Farmácia trata de todas as seções de um texto científico). Entretanto, os alunos do curso do IFSC não se sentem tão confortáveis ao usar a ferramenta quanto os alunos da Farmácia, uma vez que os exemplos da base não refletem a área em que atuam, embora tenham a mesma estrutura, como é defendido por Weissberg & Buker (1990). Esses alunos teriam mais conforto se pudessem trabalhar com textos de sua área, assim o sistema poderia indicar-lhes exemplos de textos que estão acostumados a ler. Contudo, a adaptação de ferramentas como o SciPo-Farmácia para outras áreas é custosa, pois envolve construção de *córpus* de resumos bem escritos, anotação da estrutura esquemática de cada um deles, entre outras etapas. Um processo para a construção dos recursos lingüísticos aplicáveis em ferramentas como o SciPo-Farmácia pode ser consultado em (MARQUIAFÁVEL, 2007). Além disso, seria interessante que ferramentas como o SciPo avaliassem outros aspectos além da estrutura e fornecessem *feedback* ao usuário sobre os pontos a serem melhorados, caso necessário. Dadas estas necessidades, este mestrado se propõe a estudar e avaliar soluções para a avaliação da qualidade de escrita de resumos científicos, com base nos trabalhos correlatos da literatura.

A literatura apresenta vários sistemas de avaliação automática da qualidade de escrita de dissertações (*essays*) (KUKICH, 2000; VALENTI et al., 2003; MARÍN, 2004). Esta área foi iniciada em meados da década de 60, mas somente com o desenvolvimento de outras

⁵Componentes esquemáticas podem ser refinadas em estratégias retóricas. Este refinamento é presente nas ferramentas de suporte que citamos.

⁶Usaremos os termos “texto científico” e “texto acadêmico” sem distinção.

áreas de pesquisa, principalmente de métodos estatísticos e do desenvolvimento da área de Processamento de Língua Natural (PLN), foi retomado o assunto e surgiram diversos sistemas automáticos de avaliação da escrita de dissertações, principalmente nos últimos 15 anos. Algumas dessas ferramentas alcançaram a mesma concordância com um anotador humano que a concordância medida entre dois juízes humanos. Essas ferramentas já são empregadas em avaliações reais, nas quais normalmente substituem um avaliador humano (CHODOROW; BURSTEIN, 2004).

Determinados a identificar os fatores que podem ser usados para avaliar um resumo científico, em 2004, vários pesquisadores do Instituto de Ciências Matemáticas e de Computação (ICMC) e da Northern Essex Community College, USA, iniciaram a formulação de uma rubrica para auxiliar na avaliação da qualidade de escrita de resumos científicos em inglês (ALUÍSIO et al., 2005; SCHUSTER et al., 2005). Este rubrica propõe que sejam observados sete aspectos na avaliação de resumos científicos, como organização e balanceamento da estrutura esquemática, erros gramaticais, adequação ao estilo científico, coesão, coerência, entre outros. A automatização desta rubrica permitiria que pudessem ser construídas ferramentas de auxílio à escrita capazes de informar e indicar formas do usuário melhorar seu resumo científico.

Nesse mestrado, estudamos, avaliamos e implementamos métodos de detecção automática da estrutura esquemática e de avaliação automática da qualidade de escrita de resumos científicos em inglês. Investigamos o uso de tais métodos para possibilitar o desenvolvimento de dois tipos de ferramentas: (i) de detecção de bons resumos e (ii) de crítica. O objetivo da ferramenta de detecção é discernir entre resumos bem escritos e resumos mal escritos, auxiliando na tarefa de construção de corpus para ferramentas como o SciPo-Farmácia. O objetivo da ferramenta de crítica é o de avaliar o resumo de um aluno e, se necessário, fornecer *feedback* ao aluno, indicando quais pontos devem ser melhorados. Para (i), buscamos adaptar métodos utilizados em sistemas de avaliação automática da qualidade de escrita, bem sucedidos na avaliação de dissertações e (ii) buscamos automatizar as dimensões da rubrica para resumos científicos em inglês (ALUÍSIO et al., 2005; SCHUSTER et al., 2005), para fornecer *feedback* ao usuário. Como já antecipamos, essa rubrica utiliza a estrutura esquemática do resumo na avaliação. Para tal, desenvolvemos um detector automático da estrutura esquemática de resumos em inglês.

Nossa abordagem é baseada em corpus e em aprendizado de máquina supervisionado. Utilizamos muitas técnicas de Categorização Textual (*Text Categorization*) (SEBASTIANI,

2002), embora não utilizamos a representação *bag of words*⁷ como modelagem dos documentos. Preferimos utilizar abordagens um pouco menos superficiais, aplicando ferramentas e recursos de PLN na definição e extração dos atributos.

Os atributos extraídos foram utilizados na indução de classificadores. Foram utilizados algoritmos que têm obtido bons resultados em tarefas de Categorização Textual e *Text Mining* (SEBASTIANI, 2002; GONÇALVES, 2002). Basicamente, usamos quatro algoritmos de diferentes tipos, sendo eles geradores de árvore de decisão, geradores de regras de decisão, e os estatísticos *Naive Bayes* e *Support Vector Machines* (SVM). Usamos o ambiente WEKA⁸ (*Waikato Environment for Knowledge Analysis*) (WITTEN; FRANK, 2005) para a indução e avaliação de classificadores. Para a extração dos atributos, usamos, sempre que possível, ferramentas livres disponíveis na web.

Na tarefa de detecção da estrutura esquemática, baseamo-nos nos sistemas de detecção baseados em *Argumentative Zoning* (AZ) (TEUFEL, 1999; TEUFEL; MOENS, 2002; FELTRIM, 2004) para a implementação de um detector automático da estrutura esquemática, denominado *AZEA*. Foram feitos diversos experimentos com diversas configurações de algoritmos de indução e atributos. O *AZEA* foi treinado sobre o corpus de 43 resumos do SciPo-Farmácia acrescido de 31 outros resumos, num total de 622 sentenças. Foram feitos testes com um subconjunto genérico de atributos do *AZEA*, denominado *AZSections*, para outras cinco seções dos textos científicos cobertas pelo SciPo-Farmácia.

Na avaliação da qualidade de escrita, focamos em implementações de dimensões da rubrica, sendo que implementamos as duas primeiras das sete, e no uso de atributos de outros sistemas que avaliam estilo, como o sistema de Larkey (LARKEY, 1998), além de incorporarmos também um atributo de redes complexas com fortes indícios de ligação com a qualidade de escrita (ANTIQUERA et al., 2007, 2005; ANTIQUERA, 2007). Implementamos um detector automático de erros de uso de artigo, baseado nos estudos recentes de outros sistemas com a mesma função (HAN et al., 2006; LEE, 2004). Este detector foi treinado sobre um corpus de 723 resumos científicos das áreas de Física e Ciências Farmacêuticas, construído neste mestrado. Esse detector é parte da dimensão de erros técnicos da rubrica, que não foi totalmente implementada.

Uma simples interface, na qual o usuário pode submeter seu resumo para ter detectada sua estrutura esquemática pelo *AZEA* e receber as críticas de duas das sete dimensões da

⁷Modelagem em que, basicamente, cada palavra do texto é um atributo, cujos valores podem ser a frequência da palavra no texto ou apenas a ocorrência da mesma.

⁸<http://www.cs.waikato.ac.nz/~ml/weka/index.html>

rubrica, foi feita e disponibilizada no site do NILC, denominada *AZEA-Web*⁹.

As principais contribuições deste trabalho são: (a) o desenvolvimento de um detector automático da estrutura esquemática para resumos científicos em inglês (*AZEA*) com valores de acerto e precisão acima do estado da arte; (b) exaustivos testes com diferentes algoritmos e o isolamento de problemas a serem verificados, como a confusão ao classificar sentenças das categorias Metodologia e Resultados; (c) a implementação de duas dimensões da rubrica e a implementação do detector automático de erros de uso de artigo; (d) a indicação das características dos sistemas de avaliação automática da qualidade de escrita mais pertinentes à avaliação de textos científicos. E como sub-tarefas, (e) a construção de um ferramental fácil de portar para outros sistemas ou aplicações, e com funcionalidades básicas, como o reconhecimento de tempo verbal e busca por expressões-padrão.

Embora não tenhamos construído um bom avaliador automático da qualidade de escrita, acreditamos que este trabalho indica direções para atingir esta meta, e forneça algumas das ferramentas necessárias.

Os primeiros três capítulos são de revisão da literatura. No próximo capítulo (Capítulo 2), descrevemos sistemas que auxiliam alunos na escrita de resumos científicos, fundamentados em modelos de estrutura esquemática propostos para resumos. No Capítulo 3, são definidos os conceitos, métricas e algoritmos de aprendizado de máquina utilizados neste trabalho. Também são apresentados os corpúscos usados nos treinamentos e o recursos lingüísticos e ferramentas de Processamento da Língua Natural (PLN) utilizados nas etapas de pré-processamento e extração de atributos. São apresentados três detectores automáticos da estrutura esquemática no Capítulo 4. No Capítulo 5, são apresentados os sistemas de avaliação de qualidade de escrita aplicados à dissertações (*essays*), e é apresentada e detalhada a rubrica para resumos científicos. Os experimentos realizados concentram-se nos Capítulos 6 e 7. Os experimentos, implementação e avaliação do detector automático da estrutura esquemática *AZEA* são descritos no Capítulo 6, em que também são descritos experimentos com os atributos principais do *AZEA* para outras seções de artigos, denominado *AZSections*. No Capítulo 7, são descritos as implementações e avaliações de duas dimensões da rubrica, do detector de erros de uso de artigo e um experimento com as implementações das dimensões da rubrica com atributos de sistemas e métodos descritos no Capítulo 5. Finalmente, no Capítulo 8 são apresentadas as conclusões deste trabalho e indicações de trabalhos futuros.

⁹<http://www.nilc.icmc.usp.br/azea-web/>

2 Ferramentas de suporte à escrita de textos científicos baseadas na estrutura esquemática dos textos

Buscamos neste capítulo expor as peculiaridades do gênero científico, especificamente de resumos de textos acadêmicos, as quais indicam as diferenças entre gêneros textuais e justificam a construção de ferramentas de suporte à escrita específicas para textos acadêmicos. Citamos três dessas ferramentas, e apontamos suas qualidades e limitações. Dentre as limitações, indicamos quais este trabalho pretende atacar, e usamos o contexto de ferramentas de suporte de escrita para definirmos a metodologia deste trabalho.

A Seção 2.1 introduz os trabalhos da literatura que analisam a estrutura esquemática de um resumo. Esta estrutura indica que alguns componentes são necessários, outros opcionais, e que alguns arranjos dessas componentes não são usuais e devem ser evitados para que o texto seja considerado bem escrito.

2.1 Estruturação de resumos acadêmicos

A qualidade de escrita dos resumos pode ser decisiva na divulgação científica. Devido a grande massa de publicações, muitas vezes é a leitura do resumo que implica na continuação ou não da leitura do artigo completo. Segundo Feltrim (2004), o resumo deve ser redigido com muito cuidado, de forma a ser completo (em termos das informações necessárias), interessante e informativo, dispensando a consulta ao restante do texto para que o leitor tenha a idéia do que trata o trabalho e, ao mesmo tempo, estimulando o interesse pela leitura do texto completo.

De maneira geral, os resumos podem ser de dois tipos: indicativos e informativos. Os indicativos são aqueles que apontam o que há no texto, e criam um clima de “suspense” indesejável em textos de divulgação científica. Os resumos informativos, por sua vez, são

os de preferência no meio acadêmico, pois descrevem o trabalho feito de maneira direta, antecipando os resultados alcançados.

Há vários trabalhos sobre a estrutura de resumos acadêmicos (SWALES, 1990; WEISSBERG; BUKER, 1990; ALUÍSIO; JR., 1996; FELTRIM et al., 2002). Vários autores descrevem métodos e convenções para a redação deste tipo de texto. Segundo Weissberg & Buker (1990), os resumos de pesquisa experimental de quase todas as áreas de estudo são escritos de uma maneira muito similar. Ainda sobre essa afirmação, Feltrim (2004, p. 10) adiciona que:

“Os tipos de informação incluídos e a ordem em que aparecem são muito convencionais, de modo que podem ser enunciados como modelos de resumo. Tais modelos objetivam guiar o escritor no sentido do tipo de informação que deve ser incluída em um bom resumo e da ordem que tais informações devem aparecer.”

Vários modelos de resumos têm sido propostos para informar o escritor sobre quais informações devem ser inseridas e como estas devem ser organizadas. Estes modelos mostram a estrutura esperada do resumo, denominada **estrutura esquemática**. Os modelos de estruturas esquemáticas de um resumo expõem os tipos de informação necessários, opcionais e enfatizam que essas informações obedecem à uma ordem convencional de apresentação, esperadas e observadas em resumos acadêmicos.

Na Figura 1 é apresentado o modelo de resumos de Weissberg & Buker (1990), que nomeia e descreve os tipos de informação, também denominadas componentes esquemáticas, esperados em um resumo acadêmico. Escolhemos este modelo por ser um dos mais detalhados e abranger alguns dos outros modelos citados abaixo.

Contexto	Alguma informação sobre o contexto da área de pesquisa
Propósito	A principal atividade (propósito) do estudo e seu escopo
Metodologia	Algumas informações sobre a metodologia usada no estudo
Resultados	Resultados mais importantes do estudo
Conclusão	Conclusão ou alguma recomendação

Figura 1: Modelo de um resumo típico segundo Weissberg & Buker (1990).

Na Figura 2, podemos ver a estrutura de um resumo reduzido. Nela, a informação de Contexto é descartada, além da Metodologia e o Propósito do trabalho serem escritos em uma mesma sentença. Este modelo indica quais os tipos de informação que devem ser priorizados e é ideal para revistas ou conferências que exigem resumos reduzidos.

Propósito + Metodologia
Resultados
Conclusão

Figura 2: Modelo de um resumo reduzido (WEISSBERG; BUKER, 1990).

Feltrim et al. (2002) também sugerem um modelo para a anotação de resumos, baseado em outros modelos como Swales (1990), Weissberg & Buker (1990) e Aluísio & Jr. (1996). Este modelo é composto por seis componentes esquemáticas (negrito), que são refinadas em estratégias retóricas. No modelo há três estratégias para descrever cada componente. Esse modelo é mostrado na Figura 3, sendo que a ordem a ser seguida é a mesma disposta, e as componentes marcadas com ‘*’ são obrigatórias.

-
- 1 Contexto**
 - C1. Declarar proeminência do tópico
 - C2. Familiarizar termos e conceitos
 - C3. Introduzir a pesquisa a partir da grande área
 - 2 Lacuna**
 - L1. Citar problemas/dificuldades
 - L2. Citar necessidades/requisitos
 - L3. Citar a ausência ou falta da pesquisa anterior
 - 3 Propósito***
 - P1. Indicar o propósito principal
 - P2. Detalhar/Especificar o propósito
 - P3. Introduzir mais propósitos
 - 4 Metodologia***
 - M1. Listar critérios ou condições
 - M2. Citar/Descrever materiais e métodos
 - M3. Justificar a escolha pelos materiais e métodos
 - 5 Resultado***
 - R1. Descrever os resultados
 - R2. Indicar os resultados
 - R3. Comentar/Discutir os resultados
 - 6 Conclusão**
 - C1. Apresentar conclusões
 - C2. Apresentar contribuições/valor do trabalho
 - C3. Apresentar recomendações
-

Figura 3: Componentes esquemáticas e suas respectivas estratégias retóricas (FELTRIM et al., 2002).

A proposta deste modelo foi amparada pela análise de um corpus composto por 52 resumos acadêmicos. Nesse corpus, 50% dos resumos têm entre 5 e 4 componentes, e 44,3% têm entre 3 e 2 componentes. Todos os resumos tinham a componente Propósito, e as componentes Resultados e Metodologia constam, respectivamente, em 67,3% e 63,4% dos resumos desse corpus.

Este modelo de estrutura esquemática é utilizado tanto nas ferramentas SciPo quanto no SciPo-Farmácia, embora no SciPo-Farmácia as estratégias retóricas sejam diferentes, como pode ser visto na Figura 4.

1 Contexto
C1. Declarar proeminência do tópico
C2. Familiarizar termos, objetos e processos
C3. Citar resultados de pesquisas anteriores
C4. Apresentar hipóteses
2 Lacuna
L1. Citar problemas/dificuldades
L2. Citar necessidades/requisitos
L3. Citar a ausência ou falta da pesquisa anterior
3 Propósito
P1. Apresentar o propósito principal
P2. Detalhar/Especificar o propósito
P3. Apresentar mais propósitos
P4. Apresentar o propósito com a metodologia
P5. Apresentar o propósito com os resultados
4 Metodologia
M1. Listar critérios ou condições
M2. Citar/Descrever materiais e métodos
M3. Justificar a escolha pelos materiais e métodos
5 Resultado
R1. Descrever os resultados
R2. Indicar os resultados
R3. Comentar/Discutir os resultados
6 Conclusão
C1. Apresentar conclusões
C2. Apresentar contribuições/valor da pesquisa
C3. Apresentar recomendações
C4. Apresentar lista de tópicos abordados no trabalho

Figura 4: Componentes esquemáticas e estratégias retóricas utilizadas no SciPo-Farmácia

Por último, apresentamos o modelo CARS (*Creating A Research Space*) (SWALES, 1990). O CARS, mostrado na Figura 5, foi proposto por Swales (1990) para estruturar introduções de textos científicos, mas versões modificadas deste modelo foram utilizadas para classificar resumos em um trabalho desenvolvido por Anthony & Lashkia (2003). Esse modelo foi formulado analisando-se introduções das áreas de Física, Biologia/Medicina e Ciências Sociais, e é amplamente aceito.

Movimento 1: Estabelecendo um espaço de pesquisa	
1.1	Mostrando conhecimento da área de pesquisa
1.2	Generalizando um tópico
1.3	Revisando pesquisas anteriores
Movimento 2: Estabelecendo um nicho	
2.1	Indicando uma lacuna (gap) ou
2.2	Levantando dúvidas ou
2.3	Contra-argumentando ou
2.4	Mantendo uma tradição ou
Movimento 3: Ocupando um nicho	
3.1a	Indicando os propósitos ou
3.1b	Enunciando o tópico de pesquisa
3.2	Reportando os principais resultados
3.3	Avaliando a pesquisa
3.4	Indicando a estrutura do artigo

Figura 5: Modelo CARS modificado

2.2 SciPo – *Scientific Portuguese*

O SciPo¹ é conjunto de ferramentas integradas de auxílio à escrita científica em português, resultante de um doutorado (FELTRIM, 2004). O SciPo fornece suporte à escrita das seções resumo e introdução de teses e dissertações. No SciPo, o usuário pode fornecer seu texto para que a estrutura esquemática seja detectada (composição *bottom-up*) ou compor primeiro a estrutura do texto (composição *top-down*). Quando a opção escolhida é construir primeiro a estrutura do texto, o usuário terá acesso a cinco funcionalidades de suporte que utilizam um corpúsculo anotado com a estrutura esquemática e um conjunto de regras de críticas estruturais (FELTRIM, 2004):

- (a) navegação das bases de exemplos;
- (b) pesquisa das ocorrências de determinado componente ou estratégia retórica;
- (c) apoio à composição de estruturas esquemáticas;
- (d) crítica da estrutura construída;
- (e) recuperação dos exemplos com estruturas similares à estrutura construída.

A Figura 6 diagrama as opções e o fluxo dentro do SciPo para a composição e crítica da estrutura do texto.

¹<http://www.nilc.icmc.usp.br/scipo/>

Figura 6: Visão geral do processo de estruturação do texto. Figura retirada de (FELTRIM, 2004). Na figura observamos o processo de crítica, que é cíclico, e composto dos itens (c) e (d)

O processo de construção da estrutura esquemática (c) inicia com a escolha de componentes esquemáticas e/ou estratégias retóricas, formando um primeiro esboço da estrutura esquemática. Parte-se então para o processo de crítica da estrutura (d), que é amparado por um conjunto de regras construídas com base na literatura especializada (Seção 2.1) e na análise do corpus. O processo de crítica indica desvios graves (apresentados como “críticas”) e leves (apresentados como “sugestões”). Esses desvios basicamente são referentes à necessidade ou sugestão de inserção/remoção de componentes da estrutura esquemática e da ordem de apresentação destes componentes. Um exemplo de crítica fornecido pelo sistema é mostrado na Figura 7.

Só é permitido ao usuário seguir para a redação do texto quando nenhum desvio grave for detectado. Caso contrário, o usuário continua no ciclo de modificação e crítica da estrutura esquemática.

Uma vez que o usuário compôs uma estrutura esquemática válida e está satisfeito com ela, ele pode requisitar ao SciPo que recupere textos do corpus cuja estrutura seja similar à sua (e). A similaridade é obtida pelo método de vizinho mais próximo. Os textos similares são divididos em quatro grupos, a saber:

1. contêm todas as estratégias escolhidas;

Figura 7: Crítica da estrutura. Um desvio grave foi detectado (em vermelho, falta da componente Resultados) e é sugerido que se insira as componentes Contexto e Lacuna.

2. contêm algumas das estratégias escolhidas;
3. contêm todas as estratégias escolhidas, mas em ordem distinta;
4. contêm algumas das estratégias escolhidas, mas em ordem distinta.

Como os grupos se sobrepõe, o primeiro grupo, seguindo a ordem que os descrevemos, a satisfazer as condições requeridas pelo usuário o mantém, e o texto é removido da lista dos outros grupos. Na Figura 8 é mostrada uma tela do SciPo com a recuperação de casos similares. Ao lado de cada um dos textos recuperados é dada uma medida de similaridade, que é um valor entre 0 e 100%, que estima a semelhança entre as estruturas. Esta medida é calculada distintamente para cada um dos quatro grupos, e usada no ranqueamento dos casos de um mesmo grupo.

O usuário pode analisar os textos similares, e até mesmo reutilizar a estrutura de um deles. Na redação do texto também o usuário é auxiliado pelo sistema. Dentre

Resumo - Exemplos similares

Estratégias escolhidas:

Lacuna: *Citar problemas/dificuldades*
 Contexto: *Declarar proeminência do tópico*
 Propósito: *Indicar o propósito principal*
 Resultado: *Apresentar resultados*
 Metodologia: *Citar/Descrever materiais e métodos*

[Alterar estrutura](#)

1. Exemplos que contêm todas as estratégias escolhidas:

r_es14 - 100.00%

2. Exemplos que contêm algumas das estratégias escolhidas:

r_es5 - 33.33%
 r_ic5 - 33.33%
 r_h1 - 28.57%
 r_ic1 - 28.57%
 r_sdpc3 - 28.57%
 r_ic4 - 27.27%
 r_es12 - 25.00%
 r_es3 - 25.00%
 r_h5 - 25.00%

3. Não foram encontrados exemplos que contivessem todas as estratégias escolhidas, em ordem distinta daquela previamente selecionada.

4. Exemplos que contêm algumas das estratégias escolhidas, em ordem distinta daquela previamente selecionada:

r_ic7 - 66.67%
 r_h9 - 60.00%
 r_ic6 - 50.00%
 r_h3 - 50.00%
 r_sdpc7 - 50.00%
 r_bd1 - 44.44%
 r_sdpc10 - 42.86%
 r_sdpc8 - 42.86%
 r_ic2 - 40.00%

Figura 8: Recuperação de exemplos similares, em relação à estrutura esquemática. Cada uma das listas representa um grupo de textos similares, e a estrutura construída é mostrada abaixo do título “Estratégias escolhidas”.

essas funcionalidades, podemos citar o reuso de expressões-padrão, listas de marcadores discursivos, navegação pela base de exemplos, exemplos de sentenças do corpus, entre outros. Com o texto finalizado, o usuário ainda pode salvá-lo nos formatos TXT, HTML ou RTF (*Rich Text Format*).

Além de poder compor sua estrutura esquemática desde o início, o usuário também pode submeter seu texto para que o SciPo automaticamente identifique a estrutura esquemática (abordagem *bottom-up*). Esta estrutura então é criticada, seguindo os passos que descrevemos nesta seção. A detecção automática da estrutura esquemática do SciPo é feita pelo *AZPort* (FELTRIM, 2004) e é limitada à seção resumo. O *AZPort* separa cada texto dado em sentenças e classifica cada uma delas com uma das componentes esquemáticas do modelo utilizado pelo SciPo (veja Figura 3 da Seção 2.1). A Figura 9 mostra a arquitetura do ambiente SciPo e as maneiras do usuário interagir com o ambiente, e mostra como o

AZPort está acoplado ao sistema (quadro tracejado). O usuário pode corrigir manualmente a estrutura aferida automaticamente, e então partir para a crítica da estrutura. O *AZPort* é descrito em mais detalhes no Capítulo 4, Seção 4.2.

Figura 9: Arquitetura do ambiente SciPo. Figura retirada de (FELTRIM, 2004).

2.3 SciPo-Farmácia

O SciPo-Farmácia² é um conjunto de ferramentas computacionais desenvolvido para ajudar os usuários a escreverem artigos científicos em inglês. Possui a mesma interface do SciPo, porém um número menor de funcionalidades e baseia-se em artigos científicos da área de Ciências Farmacêuticas. Este sistema foi desenvolvido com o intuito de ajudar estudantes e pesquisadores que não têm o inglês como língua materna e necessitam escrever artigos científicos nessa língua e/ou também não estão familiarizados com a estrutura e as peculiaridades do gênero científico. O desenvolvimento do SciPo-Farmácia resultou de uma parceria entre pesquisadores da Faculdade de Ciências Farmacêuticas da USP de São Paulo e o NILC.

O SciPo-Farmácia oferece apoio para o usuário compor a estrutura do texto, como é feito no SciPo (ilustrado na Figura 6). São tratadas as seis seções de um artigo científico: resumos, introduções, metodologias, resultados, discussões e conclusões. Contudo, grande parte das funcionalidades do SciPo não está presente no SciPo-Farmácia. Por exemplo, não existem regras de crítica à estrutura esquemática do texto. O usuário pode navegar pelos exemplos da base, ou mesmo requisitar exemplos similares à sua estrutura esquemática

²<http://www.nilc.icmc.usp.br/scipo-farmacia/>

na base e confrontar estruturas. Contudo, em nenhum momento o sistema indicará erros ou sugestões à estrutura. Apenas as funcionalidades mais simples, como descrição das componentes esquemáticas durante a escolha da estrutura, como pode ser visto na Figura 10, são acessíveis.

Figura 10: Exemplo de construção da estrutura esquemática para um resumo. As componentes esquemáticas e suas estratégias retóricas estão à esquerda. A janela no canto inferior direito é a resposta de uma requisição de ajuda do usuário sobre a definição da componente Conclusão.

Diferentemente do SciPo, o usuário não pode fornecer um texto já escrito, pois o SciPo-Farmácia não contém uma ferramenta para detectar a estrutura esquemática de um texto, como o *AZPort* do SciPo, que foi desenvolvido para língua portuguesa. Embora aqui tenhamos apenas apontado as diferenças do SciPo-Farmácia em comparação ao SciPo, as funcionalidades acessíveis do SciPo-Farmácia são de grande valia no auxílio à escrita de textos acadêmicos, e têm sido largamente utilizadas, por exemplo, como ferramenta de ajuda em cursos de escrita científica³.

³Há dois cursos de Pós-graduação que o utilizam regularmente: o FBC 9748-2 “Trabalhos científicos: da elaboração à publicação” (na Faculdade de Ciências Farmacêuticas da USP-SP) e o SFI 5869 “Técnicas científicas em inglês” (no Instituto de Física da USP-São Carlos)

2.4 Mover

O *Mover* (ANTHONY; LASHKIA, 2003) é um detector automático de componentes da estrutura esquemática. Destina-se a ajudar alunos não-nativos a entender a estrutura de textos científicos, associando cada sentença de um texto a categorias com funções retóricas. Foi desenvolvido em Perl, sobre o ambiente de desenvolvimento Komodo 1.2 da ActiveState⁴, para rodar no sistema operacional Windows. Esse sistema pode ser utilizado para analisar qualquer tipo de texto e não é dependente da língua em que os textos foram escritos. Usa uma abordagem de aprendizado supervisionado a partir de *cópus* anotado.

O sistema já vem treinado com um *cópus* de resumos acadêmicos anotados segundo o modelo CARS de Swales (1990) (mostrado na Figura 5), com algumas modificações. Nesse *cópus* em que foi treinado, composto por 100 *abstracts* publicados na revista *IEEE Transaction Parallel and Distributed Systems* do ano de 1998, não houve casos dos passos 1.3 (*Revisando pesquisas anteriores*), 3.1 (*Indicando os propósitos* ou *Enunciando o tópico de pesquisa*) e 3.4 (*Indicando a estrutura do artigo*) e somente um passo dos quatro apresentados na Figura 5, o 2.1 (*Indicando uma lacuna*), foi escolhido para representar o movimento de estabelecer um nicho. Restaram seis categorias que foram utilizadas pelo *Mover* para classificar sentenças:

1. *Mostrando conhecimento da área de pesquisa,*
2. *Generalizando um tópico,*
3. *Indicando uma lacuna,*
4. *Enunciando o tópico de pesquisa,*
5. *Reportando os principais resultados e*
6. *Avaliando a pesquisa.*

Os autores alegam que o sistema é de grande ajuda na leitura de resumos científicos em inglês por estudantes não nativos dessa língua. Dado um novo texto, o *Mover* segmenta-o em sentenças e classifica-as, com uma das categorias possíveis que lhe foram dadas na fase de treinamento. Essas categorias são denominadas *moves*. Podemos ver na Figura 11 um texto sendo classificado pelo *Mover* com o seu modelo próprio de treinamento.

⁴http://www.activestate.com/products/komodo_ide/. A versão atual é a 4.0.

Figura 11: Texto classificado pelo *Mover*

Os *moves* podem ser modificados, caso o usuário discorde da classificação automática. Esse texto então pode ser incorporado à base de treinamento. O *Mover* também pode ser utilizado para auxiliar na escrita. O usuário submete seu texto e tem cada uma de suas sentenças anotadas. Analisando as categorias dadas pelo sistema, o usuário pode reordenar as sentenças ou mesmo alterá-las. Nenhuma explicação sobre a estrutura é dada pelo sistema; o conhecimento dos modelos de estruturação de textos científicos fica a cargo do usuário. Em um experimento, as críticas feitas por alunos sobre seus próprios textos com a ajuda do sistema foram realizadas em tempo bem menor do que quando feitas sem o auxílio do sistema (ANTHONY; LASHKIA, 2003).

2.5 Considerações finais

Apresentamos neste capítulo modelos da literatura usados para definir a estrutura esquemática de resumos acadêmicos. Eles são fundamentados em trabalhos bem aceitos sobre a estrutura desse gênero de texto. Embora diferentes, estes modelos concordam entre si em definir os tipos de informação que devem ser desenvolvidos em um resumo acadêmico, assim como a organização desses tipos de informação no texto.

Apresentamos também as ferramentas de suporte à escrita de textos científicos que

utilizam a estrutura esquemática para aprimorar a qualidade de um texto. Apresentamos três ferramentas de suporte, e indicamos seus pontos fortes e fracos. O *Mover* é estatístico, que torna a sua transição para outros gêneros de texto (ou, por exemplo, para outras seções de textos acadêmicos) muito mais fácil, mesmo na questão da língua (embora faça segmentação de sentenças, que é dependente de língua). Entretanto, o *Mover* oferece apenas a funcionalidade de segmentação e classificação de sentenças de acordo com categorias dadas a ele durante a fase de treino, feita por meio de córpus de textos anotados com as categorias desejadas. Não oferece nenhum suporte ao usuário durante o processo de escrita.

No outro extremo temos o ambiente SciPo, que é um sistema de suporte à escrita de resumos e introduções de textos acadêmicos em português com diversas funcionalidades, tanto de suporte quanto de crítica. Ele ampara e orienta o usuário, com sugestões sobre a estrutura do texto sendo redigido e explicações sobre as convenções e componentes esquemáticas necessárias. Entretanto, muitos dos recursos utilizados no SciPo foram construídos manualmente e estão incorporados no código, como as regras de crítica, e a construção destes recursos é cara, pois demanda muito tempo de especialistas. Seguindo esta metodologia, é muito difícil portá-lo para outras línguas ou mesmo para outras seções de textos acadêmicos.

Apresentamos também o SciPo-Farmácia, que é uma ferramenta de suporte à escrita. Embora não ofereça as funcionalidades de crítica da estrutura presentes no SciPo, permite que ferramentas sejam desenvolvidas para preencher as lacunas deixadas pela ausência das ferramentas do SciPo dependentes de língua e da seção sendo tratada. Em nosso trabalho, propomos a automatização de métodos de avaliação, baseados em uma rubrica, que facilitem a portabilidade de ambientes como o SciPo-Farmácia para córpus de outra área. A formalização de recursos utilizados em ambientes como o SciPo-Farmácia foi desenvolvida em um mestrado (MARQUIAFÁVEL, 2007), e os métodos que propomos e desenvolvemos neste trabalho permitirão acelerar a construção de tais recursos. Uma vez implementadas, especialistas em uma determinada área poderão customizar um ambiente como o SciPo-Farmácia para a sua própria área de pesquisa.

3 *Conceitos de Aprendizado de Máquina, recursos e ferramentas de PLN*

Neste capítulo, definimos os conceitos e algoritmos de aprendizado de máquina e apresentamos os recursos e ferramentas de Processamento de Língua Natural (PLN) que usamos neste trabalho. Apresentamos os algoritmos mais utilizados em tarefas de Mineração de Texto (*Text Mining*) e de Categorização Textual (Text Categorization/Classification), sendo que vários destes algoritmos foram também explorados em nossos experimentos. Descrevemos também o ambiente WEKA¹ (WITTEN; FRANK, 2005), que contém diversas implementações de algoritmos de Aprendizado de Máquina, que usamos na indução dos classificadores gerados para nossos testes. As métricas utilizadas na avaliação dos classificadores são definidas e descritas na Seção 3.1.3, assim como a terminologia utilizada na avaliação. Na Seção 3.2 são descritos os recursos utilizados neste trabalho e na Seção 3.3 alguns tipos de ferramentas de PLN utilizados, sendo que para cada um desses tipos são indicadas algumas ferramentas livres disponíveis. Essas ferramentas compõem diversos passos de pré-processamento dos textos descritos nos Capítulos 6 e 7.

3.1 Indução de classificadores

Nesta seção, descrevemos alguns conceitos e métricas de avaliação de classificadores. Em especial, tratamos do aprendizado supervisionado, que prevê um conjunto de casos para os quais é sabida previamente a classe. No aprendizado supervisionado, pretende-se induzir um classificador usando os casos cuja classe é conhecida, que seja capaz de prever a classe de novos casos. Há também o aprendizado não-supervisionado (clusterização) e o semi-supervisionado. Para uma revisão da área, algoritmos e métricas veja (MITCHELL, 1997; BARANAUSKAS; MONARD, 2000; QUINLAN, 1993).

¹<http://www.cs.waikato.ac.nz/ml/weka/>

3.1.1 Conceitos de aprendizado de máquina

Nesta seção definimos os termos da área de aprendizado de máquina que utilizamos neste trabalho com o intuito de esclarecer conceitos muitas vezes ambíguos ou referenciados de várias maneiras.

Indutor: é um algoritmo (ou um programa) que, dado um conjunto de **casos** anteriormente classificados, gera um **classificador**. Este processo é denominado de “indução” ou “treinamento” de um classificador. Existem vários tipos diferentes de algoritmos que podem ser usados como indutores. Os algoritmos são separados em grupos maiores, denominados *paradigmas de aprendizado*, como, por exemplo, os algoritmos simbólicos, estatísticos, baseados em casos, conexionistas e genéticos.

Caso: um caso (também denominado exemplo ou registro) é uma lista, de comprimento fixo, de valores de atributos.

Classificador: um classificador (ou hipótese, modelo) é o produto da aplicação de um indutor sobre um conjunto de casos (também chamado conjunto de treino ou treinamento). A função de um classificador é prever corretamente a **classe** de novos casos. A qualidade do classificador depende basicamente do indutor, do conjunto de treino e da capacidade de predição dos atributos utilizados.

Classe: é um atributo especial, utilizado no aprendizado supervisionado. Em categorização textual, o termo classe é referenciado como **categoria**. Empregamos o termo *categoria* como sinônimo de **classe** neste trabalho, principalmente nas tarefas de Categorização Textual.

Classe majoritária: é a classe mais freqüente dos casos de um conjunto de treino.

Atributo: um atributo (ou *feature*) descreve alguma característica ou aspecto de um caso. O poder de predição de um atributo é uma medida subjetiva da qualidade de um atributo no auxílio do aprendizado automático de um determinado problema. Os dois tipos de atributos mais utilizados são: **nominal**, cujos valores do atributo pertencem a um conjunto finito de valores; e o **contínuo**, que é utilizado quando há uma possível ordenação nos valores. Como exemplo, temos o atributo nominal mês (jan, fev, ..., dez) e o contínuo ano (número inteiro).

Erro majoritário: é o erro de um classificador que sempre classifica uma nova entrada com a classe mais freqüente. Em suma, é 1 menos a freqüência da classe majoritária no

conjunto de treinamento.

3.1.2 Algoritmos

A vantagem de utilizarmos o ambiente WEKA para a indução de classificadores provém da facilidade do uso de diferentes algoritmos de aprendizado de máquina, uma vez extraídos os valores dos atributos e gerado um arquivo no formato ARFF (*Attribute-Relation File Format*). Com essas ferramentas pudemos utilizar os algoritmos mais empregados em tarefas de Mineração de Texto e de Categorização Textual² (JOACHIMS, 1998; MANNING; SCHÜTZE, 1999; DUMAIS et al., 1998; GONÇALVES, 2002). Nesta seção, descrevemos superficialmente cada um dos tipos de algoritmos, especificamos as implementações do WEKA utilizadas e expomos suas vantagens e desvantagens, assim como tarefas em que obtiveram bons resultados. Basicamente, foram escolhidos algoritmos de vários paradigmas de aprendizado de máquina, como o estatísticos (*Naive Bayes*, SVM), geradores de regras (RIPPER) e árvores de decisão (C4.5). Os algoritmos são detalhados a seguir.

Naive Bayes: O *Naive Bayes* é baseado na abordagem Bayesiana. Nesta abordagem, cada caso x é descrito como um vetor de valores de atributos que se deseja classificar com uma das classes de um conjunto finito V . Com a ajuda de um conjunto de treinamento, o classificador Bayesiano deve prever a classe de um novo caso descrito pelos valores de atributos $\langle a_1, a_2 \dots a_n \rangle$.

A abordagem Bayesiana para classificar esse novo caso consiste em designar a classe mais provável, v_{MAP} , dados os valores dos atributos $\langle a_1, a_2 \dots a_n \rangle$ que descrevem o caso.

$$v_{MAP} = \max_{v_j \in V} P(v_j | a_1, a_2 \dots a_n) \quad (3.1)$$

Usando o teorema de Bayes, podemos reescrever esta equação:

$$\begin{aligned} v_{MAP} &= \max_{v_j \in V} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \\ &= \max_{v_j \in V} P(a_1, a_2 \dots a_n) P(v_j) \end{aligned} \quad (3.2)$$

A determinação da probabilidade de uma classe $P(v_j)$ em um conjunto de dados é simplesmente a frequência que cada v_j ocorre no conjunto de treinamento. Entretanto,

²A representação mais utilizada nessas tarefas é a *bag of words*, que consiste em mapear a ocorrência de palavras nos documentos como atributos. Neste trabalho, entretanto, nem sempre representamos nossos casos desta maneira

estimar a probabilidade $P(a_1, a_2 \dots a_n)$ é difícil, pois o espaço de busca (possíveis combinações de valores dos atributos) é de ordem astronômica.

O classificador *Naive Bayes* é baseado na suposição de que os atributos são condicionalmente independentes, dada a classe que pretende-se estimar³. Esta simplificação permite que calculemos a probabilidade $P(a_1, a_2 \dots a_n)$ com sendo o produto das probabilidades individuais de cada atributo, ou seja, $P(a_1, a_2 \dots a_n) = \prod_i P(a_i|v_j)$. Aplicando esta simplificação à equação (3.2) temos a abordagem usada pelo *Naive Bayes*:

$$v_{NB} = \max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \quad (3.3)$$

Assumir a independência é incorreto e produz uma probabilidade incorreta dos valores de v_j para cada caso. Embora a estimativa das probabilidades seja imprecisa, o *Naive Bayes* é capaz de classificar casos com alta precisão⁴ (CRAVEN et al., 1998).

C4.5: O C4.5 (QUINLAN, 1993) é um algoritmo simbólico baseado na abordagem de árvores de decisão. Quando induzido sobre um conjunto de treino, o C4.5 gera uma árvore de decisão. Uma árvore de decisão é uma estrutura simples, onde cada nó não terminal representa testes sobre um ou mais atributos e cada nó terminal determina a decisão tomada sobre a classe a ser designada ao novo caso. O nó inicial é chamado de raiz, e os testes sempre iniciam neste nó. Na Figura 12 é mostrada um exemplo de árvore de decisão, sendo A1 e A2 atributos e “sim” e “não” os valores possíveis de classe. Se $A1 = 0$, então é classificado com “sim”, e se $A1 = 1$ e $A2 = S$, então é classificado com “não”.

Figura 12: Exemplo de árvore de decisão. O nó A1 é o nó raiz, e os nós retangulares são terminais.

³Esta é a razão do uso do termo inglês *naive*, que significa ingênuo. Tanto as grafias *naive* quanto *naïve* são corretas.

⁴Para aprofundamento na questão da suposição da independência os atributos, veja (LEWIS, 1998)

RIPPER: O algoritmo *Repeated Incremental Pruning to Produce Error Reduction* (RIPPER) (COHEN, 1995) é uma otimização do algoritmo IREP (*Incremental Reduced Error Pruning*) (FURNKRANZ; WIDMER, 1994), sendo ambos algoritmos que geram regras de decisão (QUINLAN, 1993). Dentre os algoritmos de regras, o RIPPER é um dos algoritmos com melhor performance em relação ao tempo de treinamento. As regras de decisão são da forma: **SE** <condição> $\Rightarrow$ <classe>.

Uma vez que a <condição> (que são testes sobre atributos nominais ou numéricos) é satisfeita, o caso então é identificado à <classe>. Árvores de decisão também podem ser expressas por regras, sendo que cada nó terminal da árvore gera um regra. Como são derivadas de uma árvore, cada uma das regras envolverá todos os atributos do nó raiz até cada um dos nós terminais, gerando regras mutuamente exclusivas, e portanto, podem ser aplicadas em qualquer ordem. Os algoritmos que geram regras de decisão podem gerar regras menos complexas⁵, uma vez que podem eliminar atributos desnecessários de suas regras.

Na Figura 13 é mostrado um conjunto hipotético de regras. As regras geradas pelo RIPPER são ordenadas. Dado um novo caso a ser rotulado, as regras geradas na fase de treinamento são aplicadas na ordem em que foram geradas, até que as condições de uma regra sejam satisfeitas. A última regra, que não contém condição, sempre é satisfeita. Consideremos agora um novo caso a ser classificado, descrito pelos atributos <cor=VERMELHO, comida=BOLO, tinhaMae=VERDADEIRO, folhas=FALSO >. As regras (2) e (4) são satisfeitas por este caso, mas o caso é classificado com a classe = *CARNE* pela regra (2), pois esta ocorre primeiro no conjunto de regras.

(cor = VERDE) e (folhas = FALSO)	$\Rightarrow$	classe = LEGUME	(1)
(cor = VERMELHO) e (tinhaMae = VERDADEIRO)	$\Rightarrow$	classe = CARNE	(2)
(cor = VERDE)	$\Rightarrow$	classe = VERDURA	(3)
(cor = VERMELHO) e (comida = BOLO)	$\Rightarrow$	classe = FRUTA	(4)
	$\Rightarrow$	classe = LEGUME	(5)

Figura 13: Exemplo de conjunto de regras para o RIPPER.

SVM: o *Support Vector Machine* (SVM) (VAPNIK, 1995) é muito utilizado em problemas de Mineração de Texto e Categorização Textual (JOACHIMS, 1998; GONÇALVES, 2002), principalmente quando os textos estão modelados no formato *bag of words*. A abordagem de modelos SVM baseia-se em aprendizado estatístico, combinando controle generalização com uma técnica para tratar o problema da alta dimensionalidade. O

⁵A complexidade de uma regra de decisão é medida pelo número de termos que ela consegue generalizar (QUINLAN, 1993).

SVM tem uma maior de generalização pois baseia-se no princípio de Minimização do Risco Estrutural (*Struct Risk Minimization* - SRM), e consegue induzir classificadores independentes da dimensão do espaço de atributos (JOACHIMS, 1998). Resumidamente, em exemplos de categorização binária, o SVM busca determinar um hiperplano que separe os casos positivos dos casos negativos com maior margem possível (PLATT, 1998), como ilustrado na Figura 14.

Figura 14: Exemplo de SVM linear.

3.1.3 Métricas para a avaliação de classificadores

Nesta seção, apresentamos as métricas que usamos na tentativa de estimar a performance dos classificadores que induzimos durante este trabalho. São descritos métodos de estimativa real do erro de um classificador, assim como medidas comumente usadas na avaliação, como o *Kappa* (FLEISS, 1981) e métricas para avaliação das classes, como precisão e cobertura (BAEZA-YATES; RIBEIRO-NETO, 1999).

3.1.3.1 Métodos para estimativa do erro real

Descrevemos aqui os métodos para se estimar o erro real dos classificadores que desenvolvemos neste mestrado. Embora haja outros métodos, utilizamos neste trabalho a técnica de *cross-validation*, amplamente utilizada para a validação dos resultados das métricas de avaliação. Outras maneiras de se estimar o erro real de um classificador são descritas em (BARANAUSKAS; MONARD, 2000).

- ***k-fold cross-validation***: é uma técnica bastante utilizada para estimar o erro real de um classificador. Consiste em dividir os casos aleatoriamente em k partições mutuamente exclusivas de tamanho aproximadamente de $\frac{n}{k}$, sendo n o número total de casos. Os casos das $(k - 1)$ partições são utilizados na indução de um classificador, que

é testado com a partição restante. Este processo é repetido k vezes, sempre separando uma partição diferente para teste. As métricas de avaliação são feitas calculando-se a média das métricas obtidas no teste de cada uma das k partições.

- ***k-fold stratified cross-validation***: é o *k-fold cross-validation* que procura manter a distribuição das classes (proporção de casos da classe no conjunto total de casos) em cada uma das k partições. Ou seja, se em um conjunto de 100 casos com apenas duas classes, com 70% da classe A e 30% da classe B, o *10-fold cross-validation* terá partições com 10 casos cada, sendo 7 da classe A e 3 da classe B.
- **divisão por porcentagem**: dado um conjunto de n casos e uma porcentagem p , onde $0 \leq p \leq 1$, o método *divisão por porcentagem* separa aleatoriamente o conjunto de casos em dois, sendo o de treino com $n.p$ casos e o de teste com $n.(1 - p)$.

3.1.3.2 Estatísticas de erro nas classes

Matriz de confusão: fornece detalhes sucintos dos erros e acertos de classificação cometidos por um classificador. Essa matriz confronta, para cada caso do conjunto de teste, a classe real com a classe rotulada pelo classificador. Na Figura 15, temos uma matriz de confusão genérica para k classes diferentes, $C_1, C_2, \dots, C_k$. Cada elemento m_{ij} indica o número de casos que foram rotulados com a classe C_j e são da classe C_i . Em outras palavras, o elemento m_{ij} da matriz denota “o número de casos da classe C_i que foram rotulados com a classe C_j ”.

Classe	Rotulada C_1	Rotulada C_2	...	Rotulada C_k	Total
Realmente C_1	m_{11}	m_{12}	...	m_{1k}	T_{R_1}
Realmente C_2	m_{21}	m_{22}	...	m_{2k}	T_{R_2}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Realmente C_k	m_{k1}	m_{k2}	...	m_{kk}	T_{R_k}
Total	T_{C_1}	T_{C_2}	...	T_{C_k}	N

Figura 15: Matriz de Confusão

Os totais T_{R_i} e T_{C_i} são, respectivamente, o número de casos que realmente são da classe C_i e o número de casos preditos pelo classificador como sendo da classe C_i . N é o número de casos do conjunto de teste. Com a matriz de confusão podemos obter todas as métricas descritas nesta seção, além de permitir que possamos observar mais precisamente quais pares de classes o classificador comete mais erros.

Precisão: é a razão entre as respostas corretas pelo total de respostas obtidas pelo método.

A precisão ($Prec$) de uma classe C_i é:

$$Prec(C_i) = \frac{m_{ii}}{T_{C_i}} \quad (3.4)$$

Cobertura: também referenciada como revocação ou *recall*, é a razão das respostas corretas pelo total de respostas corretas possíveis. A cobertura (Cob) de uma classe C_i é:

$$Cob(C_i) = \frac{m_{ii}}{T_{R_i}} \quad (3.5)$$

Medida-F : é uma média calculada a partir dos valores de precisão e de cobertura :

$$Medida-F(C_i) = \frac{2.Prec(C_i).Cob(C_i)}{Prec(C_i) + Cob(C_i)} \quad (3.6)$$

3.1.3.3 Estatísticas de erro geral do classificador

Taxa de acerto e erro: são as medidas mais simples e comuns na avaliação de classificadores. Possuem valores entre 0 e 1 (ou porcentagens) complementares, isto é, “*taxa de acerto + erro = 1 (100%)*”. A taxa de acerto é a razão dos casos corretamente classificados pelo total de casos classificados. Também podemos obter a taxa de acerto por meio da matriz de confusão, dada pela razão entre a soma da diagonal principal da matriz e N :

$$Taxa\ de\ Acerto = \frac{1}{N} \sum_{i=1}^k m_{ii} \quad (3.7)$$

O erro do classificador é simplesmente a diferença entre 1 e a taxa de acerto.

$$Erro = 1 - (Taxa\ de\ Acerto) \quad (3.8)$$

Macro-F: é a média aritmética das *Medidas-F* das classes. É útil para avaliar se o classificador não sacrifica o desempenho de uma ou outra classe com poucos exemplos em troca de uma melhora na taxa de acerto.

$$Macro-F = \frac{1}{k} \sum_{i=1}^k Medida-F(C_i) \quad (3.9)$$

Kappa: a estatística *Kappa* (K) é usada para medir a concordância entre as classificações distintas de N itens (FLEISS, 1981; SIEGEL; CASTELLAN, 1988). A fórmula para o cálculo do *Kappa* (K) é:

$$K = \frac{P_A - P_E}{1 - P_E} \quad (3.10)$$

em que P_A é a taxa de concordância observada e P_E estima a concordância ao acaso. O *Kappa* é uma medida de concordância com propriedades desejáveis (FLEISS, 1981). Se há total concordância, $K = +1$. Quando $K \geq 0$, a concordância observada é maior ou igual que a concordância ao acaso, fato falso quando $K < 0$. Os valores positivos do *Kappa* têm interpretações muito utilizadas em avaliações em Linguística Computacional, tanto para mensurar a concordância entre humanos quanto na avaliação de execução de uma tarefa por humanos e programas computacionais desenvolvidos para automatizar essa tarefa. A Tabela 1, proposta por Landis & Koch (1977), é comumente utilizada em Linguística Computacional, e apresenta as faixas de valores do *Kappa* com suas respectivas interpretações.

Tabela 1: Como interpretar os valores da medida *Kappa* (LANDIS; KOCH, 1977)

Valores de <i>Kappa</i>	Nível de concordância
< 0	Nenhuma
$(0 - 0, 20)$	Pouca
$[0, 20 - 0, 40)$	Mediana
$[0, 40 - 0, 60)$	Moderada
$[0, 60 - 0, 80)$	Substancial
$[0, 80 - 1]$	Quase perfeita

Para calcular o *Kappa* quando temos apenas dois classificadores, como é o caso quando comparamos a anotação humana com uma dada automaticamente, podemos usar a matriz de confusão para calcular as probabilidades P_A e P_E :

$$P_A = \frac{1}{N} \sum_{i=1}^k m_{ii} = \text{Taxa de acerto} \quad (3.11)$$

$$P_E = \frac{1}{N^2} \sum_{i=1}^k T_{C_i} \cdot T_{R_i} \quad (3.12)$$

3.1.4 O ambiente de indução de classificadores WEKA

O WEKA⁶ (*Waikato Environment for Knowledge Analysis*) (WITTEN; FRANK, 2005) é um pacote que contém diversas implementações de algoritmos de aprendizado de máquina, de

⁶A versão usada neste trabalho foi a 3.5.2. O WEKA está disponível para download no endereço <http://www.cs.waikato.ac.nz/ml/weka/>

vários paradigmas distintos. O WEKA é todo escrito em Java, o que permite portá-lo e embutí-lo em uma vasta gama de sistemas operacionais. Nesse ambiente, existem ferramentas para indução de classificadores, aplicação de classificadores para a predição da classe de novos casos e ferramentas para o cálculo de diversas métricas de avaliação dos classificadores induzidos. O WEKA trata apenas arquivos no formato ARFF, que são compostos de um cabeçalho seguidos da tabela atributo-valor, como pode ser visto na Figura 16.

<pre>@RELATION iris @ATTRIBUTE sepallength NUMERIC @ATTRIBUTE sepalwidth NUMERIC @ATTRIBUTE petallength NUMERIC @ATTRIBUTE petalwidth NUMERIC @ATTRIBUTE class Iris-setosa,Iris-versicolor,Iris-virginica</pre>
<pre>@DATA 5.1,3.5,1.4,0.2,Iris-setosa 4.9,3.0,1.4,0.2,Iris-setosa 4.7,3.2,1.3,0.2,Iris-setosa 4.6,3.1,1.5,0.2,Iris-setosa 5.0,3.6,1.4,0.2,Iris-setosa 5.4,3.9,1.7,0.4,Iris-setosa 4.6,3.4,1.4,0.3,Iris-setosa 5.0,3.4,1.5,0.2,Iris-setosa 4.4,2.9,1.4,0.2,Iris-setosa 4.9,3.1,1.5,0.1,Iris-setosa</pre>

Figura 16: Trecho de arquivo ARFF, retirado de <http://www.cs.uu.nl/docs/vakken/dm/iris.arff>.

O ARFF divide-se em duas seções: cabeçalho e dados. O cabeçalho ARFF descreve os nomes e tipos dos atributos. Os dois tipos mais comuns são mostrados na Figura 16: os nominais (*class*) e os numéricos (*NUMERIC*). A seção de dados inicia com a linha “@DATA”, seguida da tabela atributo-valor, com um caso por linha. Cada linha contém os valores de cada um dos atributos, na ordem em que foram declarados no cabeçalho. Em aprendizado supervisionado, costuma-se reservar o último atributo para a classe, embora seja possível explicitar qual atributo deve ser tratado como classe no WEKA.

O WEKA tem um ambiente visual, no qual as funcionalidades podem ser exploradas através de interação com o usuário. Na Figura 17 é mostrado o ambiente *Explorer*, utilizado no treinamento e avaliação de classificadores. Esta interface permite o usuário, de maneira amigável, excluir atributos, selecionar e alterar os parâmetros dos diferentes algoritmos de aprendizado implementados no WEKA, selecionar a maneira de avaliação (número de partições do *cross-validation*, por exemplo), entre outras opções. Ainda na Figura 17,

podemos observar as estimativas de erro, como precisão, cobertura, *Kappa*, matriz de confusão, entre outras métricas.

Figura 17: Ambiente *Explorer* do WEKA sendo utilizado para ao treinamento de classificadores

O WEKA também pode ser utilizado via linha de comando, eliminando a interação com o usuário e facilitando a integração com outros projetos. Para o treinamento, teste e avaliação de sistemas de aprendizado supervisionado, o uso do WEKA restringe-se à manipulação do seguinte comando:

```
java -cp weka.jar Indutor [Opções] [ > Resultados]
```

A seguir são descritas as variações deste comando.

Indutor é uma classe do WEKA com a implementação de algum algoritmo de aprendizado de máquina. São as classes que herdam e implementam a classe abstrata `weka.classifiers.Classifier`, como, por exemplo, a classe `weka.classifiers.bayes.NaiveBayes` (*Naive Bayes*). A coleção de algoritmos de aprendizado supervisionado é composta por 7 grupos: *bayes*, *trees*, *rules*, *functions*,

lazy, *meta* e *misc*. Alguns exemplos de indutores desses grupos são dados adiante na Seção 3.1.4.1.

Opções definem os parâmetros para treinamento, teste e saída. Além disso, parâmetros específicos de cada uma das implementações de algoritmos do WEKA podem ser configurados. Descrevemos aqui apenas as mais genéricas, usadas para o treinamento e extração das métricas que descrevemos na Seção 3.1.3:

- **-t ‘arff’**: Especifica o arquivo (‘arff’) para ser utilizado no treinamento do classificador. Quando esta opção é utilizada, são impressos os dados do classificador gerado (que dependem do indutor escolhido), seguido da avaliação. Se é dado um conjunto de teste (opção ‘-T’), é impressa a avaliação do classificador induzido sobre este conjunto de teste. Se não é usada a opção ‘-T’, são impressas as estatísticas do classificador induzido sobre o mesmo conjunto usado no treinamento e também os resultados da validação cruzada⁷(o padrão são 10 partições).
- **-T ‘arff’**: Especifica o arquivo (‘arff’) para ser utilizado como conjunto de teste, na avaliação do classificador. Se esta opção for utilizada, deve ser acompanhada ou da opção ‘-t’ ou ‘-l’.
- **-d ‘modelo’**: Salva o modelo de treinamento induzido em ‘modelo’. Este arquivo pode ser depois carregado com a opção ‘-l’, e ser utilizado para prever a classe de quaisquer conjuntos de casos que se adequem ao mesmo cabeçalho utilizado durante o treinamento.
- **-l ‘modelo’**: Carrega o modelo de treinamento salvo no arquivo ‘modelo’. Deve ser usado juntamente com a opção ‘-T’, para especificar o arquivo ARFF com os casos de teste, que devem se adequar ao mesmo cabeçalho utilizado no treinamento do modelo salvo.
- **-i**: Imprime as estatísticas de cada uma das classes, como precisão, cobertura e *medida-F*.

> **Resultados**: Os resultados apurados são impressos na saída-padrão, e podem ser redirecionados para um arquivo, com “> nome_do_arquivo”. Os resultados produzidos dependem diretamente do classificador utilizado e das opções requeridas, mas basicamente contém dados do classificador gerado (quando houver um treinamento) e

⁷*fold cross-validation*

estatísticas do treinamento ou teste (as métricas que utilizamos podem ser consultadas na Seção 3.1.3).

Por exemplo, no treinamento de um classificador *Naive Bayes* usaríamos a seguinte instrução:

```
java -cp weka.jar weka.classifiers.bayes.NaiveBayes -t treino.arff
      -d nossoClass.model -i > nossoClass.resultados,
```

e poderíamos usar esse classificador induzido em novos casos de teste, com o simples comando:

```
java -cp weka.jar weka.classifiers.bayes.NaiveBayes -T novosCasos.arff
      -l nossoClass.model -i > novosCasos.resultados
```

3.1.4.1 Indutores do WEKA

O WEKA possui várias implementações de algoritmos de diferentes paradigmas de aprendizado. Apresentamos aqui os algoritmos de aprendizado de máquina supervisionados mais utilizados em tarefas de Categorização Textual, Mineração de Texto e Detecção Automática da Estrutura Esquemática. Os algoritmos foram descritos brevemente na Seção 3.1.2, e nesta seção apenas explicitamos as implementações do WEKA para cada um destes algoritmos. Não descrevemos os parâmetros de controle específicos para cada indutor, e mantivemos sempre a configuração padrão do WEKA.

1. **NaiveBayes:** implementação do algoritmo *Naive Bayes*.

- **Paradigma:** Estatístico
- **Tarefas:** Abordagens probabilísticas como o *Naive Bayes* são reportadas entre as mais eficientes em tarefas de classificação de documentos textuais (MITCHELL, 1997; LEWIS, 1998). Além disso, também são utilizados nas tarefas de detecção automática da estrutura esquemática (TEUFEL; MOENS, 2002; FELTRIM, 2004; ANTHONY; LASHKIA, 2003).
- **Classe WEKA:** `weka.classifiers.bayes.NaiveBayes`

2. **J48:** implementação do algoritmo C4.5.

- **Paradigma:** Simbólico (Árvore de Decisão)

- **Tarefas:** Árvores de decisão, e especificamente o C4.5, são aplicados a diversos problemas de aprendizado de máquina durante os últimos 20 anos.
- **Classe WEKA:** `weka.classifiers.trees.J48`
- **Exemplo de árvore de decisão gerada**⁸

```

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

```

3. JRip: implementação do algoritmo RIPPER.

- **Paradigma:** Simbólico (geração de regras)
- **Tarefas:**
- **Classe WEKA:** `weka.classifiers.rules.JRip`
- **Exemplo de regras induzidas**⁸

```

(petallength <= 1.9) => class=Iris-setosa (50.0/0.0)
(petalwidth >= 1.7) => class=Iris-virginica (48.0/2.0)
(petallength >= 5) => class=Iris-virginica (5.0/1.0)
=> class=Iris-versicolor (47.0/0.0)

```

4. SMO: o algoritmo *Sequential Minimal Optimization* (SMO) (PLATT, 1998; KEERTHI et al., 2001) é um algoritmo baseado em SVM (VAPNIK, 1995) que otimiza o processo de treinamento do SVM padrão⁹.

- **Paradigma:** Estatístico
- **Tarefas:** Devido à sua forte capacidade de generalização, o SVM tem conquistado ótimos resultados em diversas áreas (como reconhecimento de imagens e

⁸Treinado com o arquivo completo do ARFF cujo trecho foi mostrado na Figura 16.

⁹O WEKA também possui uma classe para utilizar o classificador SVM padrão, `weka.classifiers.functions.LibSVM`. Entretanto, o LibSVM é uma implementação apenas suportada pelo WEKA, e suas bibliotecas devem ser incluídas nas classes usadas pelo WEKA. As bibliotecas do LibSVM podem ser obtidas no endereço eletrônico <http://www.cs.iastate.edu/~yasser/wlsvm/>

bioinformática), e é amplamente empregado em tarefas de Categorização Textual (JOACHIMS, 1998; DUMAIS et al., 1998; GONÇALVES, 2002).

- **Classe WEKA:** `weka.classifiers.functions.SMO`

3.2 Recursos lingüísticos para PLN: córpus e listas de frequência

Descrevemos aqui os recursos lingüísticos, anotados manualmente e disponibilizados em algum tipo de formato eletrônico, que utilizamos neste trabalho. Também são apresentados os recursos que usamos indiretamente como, por exemplo, o córpus utilizado pelos etiquetadores morfossintáticos, que influencia no conjunto de etiquetas aprendidas e usadas posteriormente na etiquetagem de novos *tokens*. Muitas ferramentas de PLN utilizam córpus em seu treinamento, e exemplos destas ferramentas serão dados posteriormente neste capítulo, na Seção 3.3. Citamos todos os córpus que usamos durante o trabalho, mas omitimos detalhes estatísticos, que apenas são apresentados nos capítulos que descrevem o uso de tais corpus.

Iniciamos descrevendo um córpus que usamos indiretamente, por meio de ferramentas de PLN treinadas sobre ele: o córpus do projeto *Penn Treebank*. O *Penn Treebank*¹⁰ (MARCUS et al., 1993) é um córpus tokenizado, anotado com etiquetas morfossintáticas, além da anotação sintática de cada sentença. Este córpus é composto por textos jornalísticos, pelos textos do córpus Brown¹¹, textos científicos, transcrições de conversas de rádio, entre outros. No total, são mais de 4,5 milhões (4.885.798) de *tokens* etiquetados morfossintaticamente, e estruturas sintáticas que englobam mais de 2,5 milhões (2.881.188) de *tokens*. As etiquetas utilizadas na etiquetagem morfossintática são mostradas no Anexo A. Este córpus é utilizado em várias ferramentas de PLN. Além disso, o seu esquema de tokenização é fornecido, e o indicamos como uma ferramenta para tokenização (Seção 3.3).

Para os experimentos para a detecção automática da estrutura esquemática, os córpus que usamos têm como base os córpus científicos do SciPo-Farmácia. No SciPo-Farmácia, existem seis córpus de diferentes seções de textos científicos (resumos, introduções, metodologias, resultados, conclusões e discussões), que juntos somam 172 textos. Estes textos foram sentenciados manualmente. Cada sentença destes textos está anotada com uma componente esquemática e estratégia retórica, dentre as possíveis do esquema de anotação

¹⁰<http://www.cis.upenn.edu/~treebank/>

¹¹O córpus Brown foi o primeiro córpus disponibilizado em formato eletrônico.

de sua seção.

Como focamos nosso estudo em resumos, ampliamos nosso *córpus* de resumos, com a adição de 31 novos resumos. Para esta ampliação, 49 resumos da área de Ciências Farmacêuticas foram obtidos da Internet (coletados do mesmo conjunto de revistas usado na coleta de resumos do SciPo-Farmácia), sentenciados manualmente e anotados quanto a sua estrutura esquemática por um especialista, seguindo o esquema de anotação do SciPo-Farmácia (as estratégias retóricas não foram anotadas). Após a anotação, o especialista dividiu os textos em dois grupos, em relação à qualidade da estruturação do resumo. Resumos cuja estrutura esquemática foi considerada atípica, isto é, incoerente com as estruturas definidas em (SWALES, 1990; WEISSBERG; BUKER, 1990), formam um *córpus* de resumos “atípicos” (18 resumos); os outros 31 resumos, como já dissemos, foram incorporados aos resumos do SciPo-Farmácia e são usados como *córpus* de treinamento para a indução de classificadores para a tarefa de detecção automática da estrutura esquemática de resumos.

Mais um *córpus*, composto de 18 resumos escritos por alunos de um curso de escrita científica, também foi sentenciado manualmente e cada uma de suas sentenças foi anotada, quanto a estrutura esquemática. Os resumos deste *córpus*, que denominamos “alunos”, foram requeridos para a matrícula no curso de escrita científica, e foram feitos pelos alunos sem nenhuma ajuda.

Esses três *córpus* (treinamento, alunos e atípicos) são detalhados no Capítulo 6. Os *córpus* de alunos e atípicos são usados apenas na avaliação do detector automático da estrutura esquemática AZEA. Os outros *córpus* do SciPo-Farmácia são utilizados também no treinamento e avaliação do classificador *AZSections*, descrito também no Capítulo 6.

Foram utilizados três *córpus* para a avaliação de qualidade de escrita, especificamente da dimensão D5 da rubrica (ALUÍSIO et al., 2005; SCHUSTER et al., 2005), desenvolvida para a avaliação da qualidade de escrita de resumos científicos em inglês, apresentada no Capítulo 5. Primeiramente, construímos um *córpus* de 723 resumos científicos das áreas de Física e Ciências Farmacêuticas, obtidos da internet de revistas de excelência. Este *córpus* apenas foi “limpo” manualmente, ou seja, trechos não pertencentes ao texto, originados na conversão de formatos, foram excluídos. Este *córpus* foi utilizado apenas para a tarefa de detecção automática de erros de uso de artigo, que faz parte da dimensão D5 da rubrica para resumos, e é descrito no Capítulo 7.

Os outros dois *córpus* foram obtidos dos pesquisadores envolvidos na definição da rubrica para a avaliação da qualidade de escrita de resumos científicos em inglês. Na verdade,

é um *córpus* composto por 78 pares de resumos. Cada par de textos corresponde a um resumo escrito pelo estudante, como requisito para ingressar em um curso de escrita científica, e uma versão corrigida por um lingüista nativo da língua inglesa. Os resumos foram escritos por estudantes de pós-graduação das áreas de Física e Ciências Farmacêuticas. Além disso, os comentários da correção também estão anotados, possibilitando que erros sejam isolados. Dividimos os pares de resumos e os separamos em dois *córpus*: resumos “bons” (corrigidos) e resumos “ruins” (primeiro resumo dos alunos). Embora todos os resumos tenham pelo menos uma correção, apenas um especialista anotou estes textos, e não temos, por exemplo, o *Kappa* desta tarefa para justificarmos formalmente a divisão dos resumos nestes dois grupos.

Para a tarefa de detecção automática de erros de uso do artigo (Capítulo 7), foi usada a lista de frequência do *córpus* BNC¹² (*British National Corpus*) para auxiliar na calibragem do atributo *Countability*. Não utilizamos os textos do *córpus* BNC, apenas a lista de frequência dos *tokens* de seu *córpus*. A lista que utilizamos foi construída por Adam Kilgariff, e está disponível para download¹³.

3.3 Ferramentas de PLN

O Processamento de Língua Natural (PLN) é um ramo da Inteligência Artificial que estuda os problemas inerentes à manipulação da língua natural. Chamamos aqui de ferramentas de PLN ou ferramentas lingüísticas os programas de computador que recebem um texto (ou um trecho de texto) em língua natural e agregam ou extraem algum tipo de informação ou traço lingüístico.

As ferramentas de PLN são normalmente encadeadas umas nas outras, normalmente recebem como entrada a saída de outra ferramentas lingüística, efetuam sua parcela de processamento, e produzem uma saída que poderá servir como entrada para outra ferramenta lingüística. Embora esse encadeamento seja natural, pois muitas vezes as análises lingüísticas manuais seguem o mesmo processo, erros no início do processo são propagados para as ferramentas seguintes, sendo que pequenos erros podem resultar em desastres ao fim do processo. Dessa maneira, embora muitas tarefas sejam simples, as ferramentas que automatizam estas tarefas devem ser mais precisas quanto mais inicialmente são aplicadas.

Definimos os tipos de ferramentas para as tarefas mais basilares, que tratam informações do nível lexical e sintático, e damos exemplos de ferramentas disponíveis.

¹²<http://www.natcorp.ox.ac.uk/>

¹³<http://www.kilgariff.co.uk/bnc-readme.html>

Sempre que possível, indicamos a precisão da tarefa e também se há alguma ferramenta equivalente para o português. Por fim, mostramos um pacote de ferramentas de PLN com a implementação de várias delas para o inglês, denominado *OpenNLP*. Comecemos primeiro apontando recursos lingüísticos utilizados em ferramentas baseadas em corpus, tanto que usamos quanto as que propomos.

3.3.1 Sentenciador

Neste trabalho, definimos um sentenciador por sua função. A função de um sentenciador é indicar onde acaba uma sentença e inicia-se outra. Embora pareça uma tarefa trivial, sentenciadores são quase sempre as primeiras ferramentas lingüísticas aplicadas em um pré-processamento, e quebras de sentença erradas podem gerar sentenças agramaticais, e ampliar o erro das ferramentas posteriores. Sentenciadores devem ser capazes de reconhecer quando um “.” (ponto) é final, ou se apenas faz parte de uma abreviatura ou de uma notação numérica, por exemplo. O conhecimento de tais formações normalmente ataca o sentenciador a uma língua específica.

Existem vários sentenciadores livres para uso. Entre os disponíveis, destacamos dois deles. O *SENDER*¹⁴, desenvolvido no NILC, tem versões para português e inglês, ambas para o sistema operacional Windows. O segundo sentenciador compõe o pacote *OpenNLP*¹⁵, que descrevemos mais adiante.

3.3.2 Tokenisador

Um tokenisador deve determinar o que será tratado como *token*¹⁶. Assim como o sentenciador, um bom tokenisador deve lidar com abreviações, números, e outras desambiguações que dependem de conhecimento da língua do texto de entrada.

Um simples tokenisador é fornecido pelo projeto *Penn Treebank*. Na página em que é descrita a tokenização empregada no projeto¹⁷ existe um *script sed* para a tokenização¹⁸. A entrada para esse tokenisador deve estar separada, uma sentença por linha.

¹⁴<http://www.icmc.usp.br/~tasparado/Senter.htm>

¹⁵<http://opennlp.sourceforge.net/>

¹⁶Em computação, *token* é o menor bloco estruturado de um texto. Em nosso trabalho, são palavras e pontuações (que compreendem símbolos tais como “;”, “:”, “(”, ...), uma definição que normalmente é utilizada. Mas sua definição pode ser estendida ou alterada, desde que segmente o texto em blocos úteis para uma tarefa qualquer.

¹⁷<http://www.cis.upenn.edu/~treebank/tokenization.html>

¹⁸<http://www.cis.upenn.edu/~treebank/tokenizer.sed>

3.3.3 Etiketador Morfossintático

Também denominado de *part of speech tagger* ou simplesmente *tagger*, sua função é agregar informação morfossintática (*Part Of Speech* ou *POS*) a cada *token* de um texto. A informação agregada é denominada “*etiqueta*”. Exemplos de etiquetas morfossintáticas são: *substantivo*, *adjetivo*, *verbo*, *conjunção*, entre outros.

Existem várias abordagens para a realização da etiquetagem morfossintática. Basicamente, todas as abordagens dependem da língua (mesmo que indiretamente, no caso de etiquetadores que são baseados em córpus, e portanto o córpus deve ser da língua-alvo) e do conjunto de etiquetas adotado, que deve ser finito. A grande maioria é baseada em aprendizado de córpus, em que cada *token* dos textos é anotado manualmente com uma etiqueta. A taxa de acerto dos etiquetadores mais usados (e disponíveis para obtenção e uso) para a língua inglesa é superior a 96%. Entre os etiquetadores mais conhecidos e usados, temos o TBL (*Transformation-Based Learning*), criado por Eric Brill¹⁹ que é baseado em regras, que são extraídas de córpus, e os etiquetadores estatísticos baseados em córpus TreeTagger²⁰ e o MXPOST²¹.

O **MXPOST** (RATNAPARKHI, 1996) é um etiquetador morfossintático estatístico baseado em máxima entropia (RATNAPARKHI, 1997), juntamente com vários atributos contextuais. O MXPOST é também baseado em aprendizado de córpus, e precisa ser previamente treinado a partir de um córpus anotado com as etiquetas que se pretende prever. Em textos jornalísticos em inglês, o MXPOST reporta o erro de 3,4% (RATNAPARKHI, 1996). Ele é composto por funções de treinamento e uso de etiquetadores, escrita em Java. No pacote disponível para download²², juntamente com o MXPOST está um etiquetador, previamente treinado com o córpus *Wall Street Journal* do projeto *Penn Treebank*²³ (MARCUS et al., 1993). O MXPOST assume que o texto de entrada está separado em sentenças (uma por linha) e tokenizado (*tokens* separados por espaço). Na Figura 18, temos um texto de quatro sentenças (este texto foi primeiramente sentenciado e tokenizado) pronto para ser etiquetado pelo MXPOST, seguido do mesmo texto já etiquetado.

O conjunto completo das etiquetas utilizadas no *Penn Treebank* (e, conseqüentemente, usado pelo MXPOST) pode ser visto no Anexo A.

¹⁹<http://www.cs.jhu.edu/~brill/>

²⁰<http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/>

²¹<http://www.cogsci.ed.ac.uk/~jamesc/taggers/MXPOST.html>

²²<ftp://ftp.cis.upenn.edu/pub/adwait/jmx/jmx.tar.gz>

²³<http://www.cis.upenn.edu/~treebank/>

<p>The investigation , into the conditions in the sociology department , will be conducted by a commission of faculty members and administrators that was formed last week .</p> <p>It follows a rare and remarkable burst of defiance and student activism on a Russian campus , a case of grass-roots organization and civic activity that private groups and critics of the Kremlin have said has been in decline in recent years .</p>
<p>The DT investigation NN , , into IN the DT conditions NNS in IN the DT sociology NN department NN , , will MD be VB conducted VBN by IN a DT commission NN of IN faculty NN members NNS and CC administrators NNS that WDT was VBD formed VBN last JJ week NN . .</p> <p>It PRP follows VBZ a DT rare JJ and CC remarkable JJ burst NN of IN defiance NN and CC student NN activism NN on IN a DT Russian JJ campus NN , , a DT case NN of IN grass-roots JJ organization NN and CC civic JJ activity NN that IN private JJ groups NNS and CC critics NNS of IN the DT Kremlin NNP have VBP said VBN has VBZ been VBN in IN decline NN in IN recent JJ years NNS . .</p>

Figura 18: Entrada e saída do etiquetador morfossintático MXPOST. Separados por espaços estão os pares *token-etiqueta*, sendo que as etiquetas morfossintáticas estão em negrito. Trecho retirado do jornal *NY Times*(<http://www.nytimes.com/2007/03/22/world/europe/22russia.html>)

Para o português, foi conduzido um estudo da aplicação de etiquetadores morfossintáticos baseados em corpus do português (brasileiro) (AIRES, 2000), com aproximadamente 100 mil palavras. Foram treinados posteriormente três etiquetadores morfossintáticos, com um corpus de 1 milhão de palavras, que pode ser obtidos no site do NILC²⁴. O treinamento e a disponibilização estão engajados no projeto Lácio-Web²⁵, também do NILC.

3.3.4 *Chunker*

Um *chunker* segmenta uma sentença em unidades estruturais denominadas *chunks*. Um *chunk* é um conjunto de *tokens* consecutivos, agrupados por função sintática, como sintagmas nominais (*Noun Phrases* ou NP), verbais (VP) e preposicionais (PP). Os tipos (funções sintáticas) possíveis de *chunks* que utilizamos neste trabalho são descritos em Sang &

²⁴<http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html>

²⁵<http://www.nilc.icmc.usp.br/lacioweb/>

Buchholz (2000). Os *chunks* não se sobrepõem (nunca dois *chunks* agrupam um mesmo *token*) e não são hierárquicos (não há *chunks* de *chunks*, como em um *parser*). Normalmente, a entrada de um *chunker* deve ser um texto tokenizado e etiquetado morfossintaticamente. Na Figura 19 é exibida uma sentença separada em *chunks*, pelo o *chunker* do conjunto de ferramentas *OpenNLP*. As etiquetas morfossintáticas foram omitidas, apenas para facilitar a visualização.

[_{NP} I] [_{VP} was carefully picking] [_{PRT} up] [_{NP} the snake]
 [_{ADVP} when] [_{NP} it] [_{VP} bit] [_{NP} me] .

Figura 19: Uma sentença separada em *chunks*. Os *chunks* são delimitados por ‘|’ e ‘|’, e suas funções sintáticas, que são apontadas no início do *chunk*. ADVP é um sintagma adverbial e PRT é uma partícula, sendo que o conjunto de etiquetas é descrito em (MARCUS et al., 1993).

3.3.5 Ferramentas lingüísticas do pacote *OpenNLP*

O pacote *OpenNLP* é uma coleção de ferramentas de PLN escrita em Java²⁶. Para o inglês, existem implementações de sentenciador, tokenizador, etiquetador morfossintático, *chunker* e *parser*. As ferramentas disponíveis para a língua inglesa estão no pacote `opennlp.tools.lang.english`.

O *tagger* contido no pacote de ferramentas *OpenNLP* (`opennlp.tools.lang.english.PostTagger`) também é baseado em Máxima Entropia, e foi treinado com o corpus *Penn Treebank*, descrito na Seção 3.2. Sua precisão é superior a 96%.

Na Figura 20, descrevemos um *script* em *bash* que recebe um texto qualquer em língua inglesa e aplica todas as ferramentas necessárias para se obter um texto separados em sentenças e com os *tokens* etiquetados morfossintaticamente e agrupados em *chunks*.

3.3.6 Conversores de formatos

Para grande parte das ferramentas lingüísticas, a entrada deve ser feita por meio de arquivos de “texto puro” (TXT), ou seja, sem outras informações, como etiquetas (em arquivos HTML) ou outras informações de apresentação e formato (como documentos PDF, por exemplo). Apresentamos aqui conversores dos formatos mais utilizados no armazenamento e disposição de informação textual. Escolhemos apenas conversores não-comerciais, e buscamos, a princípio, sistemas disponíveis para vários sistemas operacionais (focando principalmente

²⁶Disponível para obtenção e uso em <http://opennlp.sourceforge.net/>

```
#!/bin/bash

LIB=$AED_DIR/sources/jars/    #diretórios dos jars
MODEL="$LIB/models" #diretórios dos modelos treinados

#arrumando classpath...
JARS="$CLASSPATH:.$LIB/opennlp-tools-1.3.0.jar"
JARS="$JARS:$LIB/maxent-2.4.0.jar:$LIB/trove.jar:$LIB/jwnl-1.3.3.jar"
export CLASSPATH=$JARS

#chunker:sentencia, tokenisa e segmenta em chunks. Resultado na saída-padrão
cat $1 |
  java opennlp.tools.lang.english.SentenceDetector $MODEL/EnglishSD.bin.gz |
  java opennlp.tools.lang.english.Tokenizer $MODEL/EnglishTok.bin.gz |
  java opennlp.tools.lang.english.TreebankChunker $MODEL/EnglishChunk.bin.gz
```

Figura 20: *Script* com aplicação do sentenciador, do tokenizador e do *chunker* do pacote *OpenNLP* para a extração dos *chunks* de um texto qualquer em inglês.

nos sistemas Linux e Windows). São apresentados três conversores, que cobrem os formatos PDF, DOC (MSWord e OpenOffice) e páginas Web (basicamente HTML, mas engloba também alguns outros formatos). Todos geram arquivos TXT. Outro critério considerado foi a facilidade de acoplagem do sistema em outros (normalmente a conversão é a primeira etapa do pré-processamento, quando utilizada), e apenas sistemas sem interação direta com o usuário²⁷ durante a conversão são considerados.

- **XPDF:** é um software que agrega várias funções de visualização e conversão do formato PDF para outros formatos. Em particular, o XPDF contém um extrator de texto PDF, `pdftotext`, que converte documentos PDF para TXT.

Link: <http://www.foolabs.com/xpdf/>

Sintaxe: `pdftotext -raw texto_entrada.pdf texto_saida.txt`

Problemas: Quando o texto está contido em imagens ou em arquivos encriptados, a extração é impossibilitada ou prejudicada. Nesses casos, normalmente, ou nenhum texto é extraído ou é extraído “*lixo*”²⁸.

- **AbiWord:** é um editor de texto similar ao Microsoft Word®. Pode ser utilizado para converter formatos tipicamente manuseados pelo Word®, como DOC e RTF (*Rich Text*

²⁷Sem interação direta consiste apenas na possibilidade de execução de um processamento por meio de comandos previamente construídos. Interação direta implica que o usuário necessita, em tempo de execução, inserir comandos ou efetuar ações (com o mouse, por exemplo).

²⁸seqüência de caracteres sem sentido

Format) para TXT. Foi desenvolvido para suportar os arquivos gerados pelo *Microsoft Word*®, *OpenOffice*, *Word Perfect*, entre outros. Alguns outros pacotes também tem ferramentas de conversão de formatos, como o Antiword²⁹ e o wvWare³⁰.

Link: <http://www.abisource.com/>

Sintaxe: `abiword -to=txt texto_entrada.doc` Texto convertido será salvo em `texto_entrada.txt`.

Problemas: Tabelas tendem a gerar muito lixo. A saída privilegia a visualização, e as vezes mantém espaçamentos desnecessários.

- **Lynx:** é um navegador de modo texto para páginas WWW (*World Wide Web*). O *Lynx* não só converte arquivos HTML, por exemplo, mas também extrai o texto de uma URL (*Universal Resource Locator*). Isto permite, por exemplo, que páginas PHP sejam processadas pelo servidor e então o texto da página exibida seja extraído. Existem outros sistemas com mesma funcionalidade, como o *Elinks*³¹.

Link: <http://lynx.browser.org/>

Sintaxe: `lynx URL -dump URL` é o endereço da página a ser convertida.

Problemas: A extração de textos de páginas de internet é acompanhada de muito lixo. Normalmente páginas estão cheias de *links* para outras páginas, a visualização é muito importante para documentos dispostos na *web*. Outro problema são as páginas com *frames*, cujo conteúdo não está na URL principal, e sim em outras páginas. Este mesmo problema ocorre em páginas com redirecionamento.

3.4 Considerações finais

Neste capítulo definimos os conceitos de Aprendizado de Máquina da abordagem supervisionada, que utilizamos nos próximos capítulos. Descrevemos sucintamente os algoritmos mais utilizados em áreas correlatas às que estudamos neste trabalho, sendo que muitos serão usados nos capítulos seguintes, e explicitamos as implementações que foram usadas quando necessária a indução de um classificador. Apresentamos os recursos lingüísticos, na maioria córpis, que usamos neste trabalho ou que são usados por ferramentas

²⁹<http://www.winfield.demon.nl/>

³⁰<http://wvware.sourceforge.net/>. Nessa página, entretanto, os mantenedores do wvWare sugerem que sejam usadas as ferramentas de conversão do AbiWord

³¹<http://elinks.or.cz/>

de PLN basilares. Por fim, apresentamos as ferramentas de PLN utilizadas em várias das etapas de pré-processamento para a modelagem e representação dos textos em língua natural que tratamos. São fornecidas fontes de onde se encontrar tais ferramentas, sendo que as mais importantes em nosso trabalho são exemplificadas com mais detalhes.

4 *Métodos para detecção automática da estrutura esquemática de textos científicos*

Neste capítulo, apresentamos três métodos de detecção automática da estrutura esquemática de textos científicos. O primeiro deles, o **AZ** (*Argumentative Zoner*, Seção 4.1), foi desenvolvido para a determinação de autoria das sentenças de artigos científicos em inglês¹. O segundo, o **AZPort** (Seção 4.2), adaptou o método *AZ* para resumos de textos acadêmicos escritos em português. Por último, descrevemos o método usado na ferramenta de suporte à escrita *Mover* (Seção 4.3), que é independente de língua e que foi aplicado a textos científicos em inglês com o mesmo propósito do *AZPort*. É apresentada também a avaliação intrínseca para cada um dos métodos, da forma como foi relatada por seus respectivos autores.

4.1 *Argumentative Zoning*

Argumentative Zones (zonas argumentativas) (TEUFEL, 1999; TEUFEL; MOENS, 2002) são segmentos com função retórica específica dentro de um texto completo. Foi proposto inicialmente (em (TEUFEL, 1999)) para identificar a autoria de cada sentença de textos científicos, que podiam ser rotuladas como: informações de contexto (ou senso comum da área), pesquisa/contribuição do próprio autor ou pesquisa/contribuição de outros autores. A detecção automática dessas zonas argumentativas, denominada *Argumentative Zoner*, que abreviaremos aqui por **AZ**, é feita por meio do treinamento de um classificador usando o algoritmo estatístico de aprendizado de máquina Naive Bayes (LEWIS, 1998). Para o treinamento, cada sentença é processada para apurar o valor de 16 atributos, que extraem características lexicais, sintáticas e estruturais. É importante ressaltar que esse classificador supõe que os atributos sejam estatisticamente independentes, pois isso influenciou na

¹Esta tarefa não é a de identificar textos de diferentes autores. Consiste em determinar, entre as sentenças de um mesmo artigo, quais relatam trabalho atual do próprio autor e quais relatam trabalhos prévios.

definição e escolha dos atributos utilizados no *AZ*. O *AZ* foi posteriormente adaptado para a sumarização de textos científicos, selecionando material para o sumário que destacava as novas contribuições do artigo fonte e o situava em relação a trabalhos prévios (TEUFEL; MOENS, 2002). Para tal tarefa, cada sentença do texto foi classificada em sete categorias retóricas, que podem ser vistas na Tabela 2. Os 16 atributos utilizados por Teufel & Moens (2002) para a determinação automática da categoria de uma sentença podem ser vistos na Tabela 3. De fato, é o mesmo conjunto de atributos usados em (TEUFEL, 1999), com algumas alterações nos possíveis valores dos atributos.

Tabela 2: Esquema de anotação do *AZ* (TEUFEL; MOENS, 2002).

Categoria Retórica	Descrição
<i>Background</i>	Fatos e conhecimentos geralmente aceitos
<i>Other</i>	Descrição do trabalho de outras pessoas
<i>Own</i>	Descrição do próprio trabalho próprio: metodologia, resultados, discussão
<i>Aim</i>	Propósito da pesquisa atual
<i>Textual</i>	Seção de estrutura/organização do texto
<i>Contrast</i>	Contraste e comparação com outros trabalhos, lacunas de outros trabalhos
<i>Basis</i>	Outro trabalho que serve de base para este e concordância com outros trabalhos

4.1.1 Avaliação Intrínseca

Na avaliação feita em Teufel et al. (1999) foi utilizado um *cópus* anotado de 48 artigos em inglês da área de *Linguística Computacional*. Primeiramente, foi levantada a reprodutibilidade do sistema, medida por meio da concordância entre anotadores humanos. Três anotadores humanos treinados classificaram 4261 sentenças dos artigos do *cópus* com uma das sete categorias do *AZ* (Tabela 2). O valor do *Kappa* (o cálculo deste valor é descrito na Seção 3.1.3) obtido foi de 0,71, que indica que o esquema de anotação proposto é reproduzível. Para a avaliação do sistema, primeiramente foram computados os valores dos atributos (descritos na Tabela 3) de cada sentença do *cópus*. Com esses valores foi treinado o *AZ*, que foi avaliado usando o *10-fold stratified cross-validation* e seus resultados foram comparados com três *baselines*, mostrados na Tabela 4.

Na Tabela 4, como *baseline I* foi utilizada a saída do classificador textual *RainBow* (apud (TEUFEL; MOENS, 2002)), que classifica a sentença de acordo com a medida $TF \cdot IDF^2$;

² *Term frequency * Inverse Document Frequency*

Tabela 3: Atributos utilizados pelo AZ (TEUFEL; MOENS, 2002)

Nome	Descrição	Valores
<i>Location</i>	Posição da sentença	A-J (10 posições)
<i>Section structure</i>	Posição da sentença dentro de uma seção	7 possíveis posições
<i>Paragraph structure</i>	Posição relativa da sentença dentro de um parágrafo	<i>Initial, Medial, Final</i>
<i>Headline</i>	Tipo do título da seção atual	15 tipos pré-estabelecidos ou <i>Non-Prototypical</i>
<i>Length</i>	A sentença, em número de palavras, é longa ou curta?	Sim ou Não
<i>Title</i>	A sentença contém palavras também contidas no título ou no subtítulo?	Sim ou Não
<i>TF*IDF</i>	A sentença contém termos significativos, segundo a medida TF*IDF?	Sim ou Não
<i>Voice</i>	Voz do primeiro verbo finito da sentença	<i>Active, Passive</i> ou <i>NoVerb</i>
<i>Tense</i>	Tempo do primeiro verbo finito da sentença	9 tempos verbais (simples e complexos) ou <i>NoVerb</i>
<i>Modal</i>	O primeiro verbo finito da sentença modificado por um auxiliar modal?	<i>Modal, NoModal</i> ou <i>NoVerb</i>
<i>Citation</i>	A sentença contém uma citação ou o nome de autor da lista de referências? Se sim, é uma citação de si mesmo? Em que posição da sentença a citação ocorre?	<i>Citation (self), Citation (other), Author Name</i> , ou <i>None X Beginning, Middle, End</i>
<i>History</i>	Categoria provável da sentença anterior	7 categorias + ‘ <i>Begin</i> ’
<i>Formulaic</i>	Tipo da expressão-padrão presente na sentença	18 tipos de grupos de expressões-padrão + 9 tipos de Agente ou <i>None</i>
<i>Agent</i>	Tipo do agente	9 tipos de Agente ou <i>None</i>
<i>Seg Agent</i>	Tipo do agente da sentença anterior	9 tipos de Agente ou <i>None</i>
<i>Action</i>	Tipo da ação, com ou sem negação	27 tipos de ação ou <i>None</i>

a *baseline II* atribuiu aleatoriamente uma categoria à sentença; a *baseline III* também atribuiu uma categoria à sentença aleatoriamente, mas utiliza a distribuição das categorias do corpus; e a *baseline IV* sempre atribui a categoria mais freqüente à sentença.

4.2 *Argumentative Zoning for Portuguese (AZPort)*

Em um doutorado desenvolvido no ICMC, Feltrim (2004) propôs e implementou um detector automático da estrutura esquemática de resumos de textos científicos em português,

Tabela 4: Avaliação geral do *AZ* (TEUFEL; MOENS, 2002)

	<i>AZ</i>	<i>baseline I</i>	<i>baseline II</i>	<i>baseline III</i>	<i>baseline IV</i>
<i>Kappa</i>	0,45	0,30	-0,10	0	-0,13
Taxa de Acerto	0,73	0,72	0,14	0,48	0,67
<i>Macro-F</i>	0,50	0,30	0,09	0,14	0,11

denominado *AZPort*. Esse trabalho foi baseado no método de *Argumentative Zoning* (TEUFEL, 1999; TEUFEL; MOENS, 2002), descrito na Seção 4.1. No *AZPort*, cada sentença pode ser de um dos sete tipos apresentados na Tabela 5. Para realizar a classificação, o *AZPort* utiliza 8 atributos, que estão resumidos na Tabela 6, e que são utilizados no treinamento usando o algoritmo *Naive Bayes*.

Tabela 5: Classes possíveis para cada sentença no *AZPort*

Categoria Retórica	Descrição
Contexto	Fornece um contexto para o problema a ser analisado
Lacuna	Indica a necessidade de mais investigação na área
Propósito	Propósito ou objetivo do trabalho
Metodologia	Metodologia utilizada na realização do trabalho
Resultado	Descreve os principais resultados do trabalho
Conclusão	Implicações e sugestões de continuação e aplicação do trabalho
Estrutura	Seção de estrutura/organização do texto

4.2.1 Avaliação Intrínseca

Os resultados foram computados pelo autor aplicando-se o *13-fold cross-validation* a 52 resumos em português. Na comparação, foram utilizadas duas *baselines*. A primeira escolhia aleatoriamente a categoria da sentença, e a segunda atribuía a categoria mais freqüente a todas as sentenças. As métricas utilizadas na comparação foram a medida *Kappa* e a taxa de acerto, cujos valores são mostrados na Tabela 7.

O valor do *Kappa* é melhor do que o reportado em (TEUFEL; MOENS, 2002) (*Kappa* de 0,45), que segundo o autor pode ser devido ao recorte, pois o *AZPort* foi aplicado à resumos da área de Ciências da Computação, enquanto o *AZ* foi aplicado a artigos científicos de lingüística computacional. Ainda segundo o autor, o atributo que mais contribui para o desempenho do classificador é o atributo *Expressão*, que é o atributo com maior poder de distinção, seguido do atributo *Histórico*.

Tabela 6: Resumo dos atributos utilizados pelo *AZPort* na classificação de sentenças de resumos

Atributo	Descrição	Valores Possíveis
Tamanho	Tamanho da sentença	curta, média ou longa
Localização	Posição relativa da sentença no resumo	primeira, segunda, mediana, penúltima ou última
Citação	Presença de citações na sentença	sim ou não
Expressão	Presença de expressões-padrão	contexto, lacuna, propósito, metodologia, resultado, conclusão ou <i>noexpr</i>
Tempo	Tempo do primeiro verbo finito da sentença	IMP, PRES, PAST, FUT, PRES-CPO, PASTCPO, FUT-CPO, PRES-CT, PAST-CT, FUTCT, PRES-CPO-CT, PAST-CPO-CT, FUTCPO-CT ou <i>noverb</i>
Voz	Voz do primeiro verbo finito da sentença	Passiva, ativa ou <i>noverb</i>
Modal	Se um verbo é ou não modal	sim, não ou <i>noverb</i>
Histórico	Categoria da sentença anterior	contexto, lacuna, propósito, metodologia, resultado, conclusão, estrutura ou “início”

Tabela 7: Avaliação do *AZPort*

	<i>AZPort</i>	<i>baseline I</i>	<i>baseline II</i>
<i>Kappa</i>	0,65	0	0,26
<i>Taxa de acerto</i>	72%	20%	32%

4.3 O analisador automático do *Mover*

O *Mover*³ (ANTHONY; LASHKIA, 2003) é uma ferramenta que utiliza um detector automático das componentes da estrutura esquemática. O *Mover* pode ser utilizado para analisar qualquer tipo de texto e não é dependente da língua nem do domínio em que os textos foram escritos. Destina-se a ajudar alunos, especialmente não-nativos, a entender a estrutura de textos científicos, associando cada sentença a uma componente esquemática.

O *Mover* já vem treinado para a classificação de sentenças de resumos. O esquema de anotação, relatado na Seção 2.4 do Capítulo 2, é uma modificação do modelo CARS (*Creating A Research Space*) de Swales (1990) (apresentado na Figura 5 do Capítulo 2. O esquema de anotação, mostrado na Figura 21, é composto por seis componentes, que são

³<http://www.antlab.sci.waseda.ac.jp/software.html>

chamadas de *moves*.

Componente esquemática (<i>move</i>)	Movimento em (SWALES, 1990)
M ostrando conhecimento da área de pesquisa	1.1
G eneralizando um tópico	1.2
I ndicando uma lacuna (gap)	2.1
E nunciando o tópico de pesquisa	3.1b
R eportando os principais resultados	3.2
A valiando a pesquisa	3.3

Figura 21: Esquema de anotação usado no *Mover*

O *Mover* implementa um *Naive Bayes* (LEWIS, 1998). Os atributos de classificação baseiam-se no modelo *bag of words*, embora utilize expressões de 1 a 5 palavras em vez de palavras simples. Essa adaptação foi denominada de *bag of clusters*. Dessa maneira, cada sentença é separada em *clusters*, e cada *cluster* é mapeado na categoria da sentença a qual ele pertence. Também é utilizada uma medida estatística chamada *Information Gain* (IG) (MANNING; SCHÜTZE, 1999), que mede a importância dos *clusters* na diferenciação de categorias e também é utilizada para remover “ruídos”, que são *clusters* com pouca importância na classificação. Desse modo, é possível utilizar-se da medida IG para o ranqueamento e escolher um limiar de corte, que idealmente extinguiria o ruído. O *Mover* também utiliza outros atributos, como posição da sentença no texto e probabilidades de fluxo de categorias, semelhante ao atributo *Histórico* do *AZPort*, descrito anteriormente. Na fase de teste, o *Mover* segmenta o texto em sentenças e rotula cada uma delas com um de seus *moves* possíveis dados no treinamento.

4.3.1 Avaliação Intrínseca

O *Mover* foi treinado com 554 sentenças e testado sobre 138 sentenças. Foram gerados e testados cinco conjuntos de teste e treino randomicamente, o que resultou numa taxa de acerto do *Mover* de 68%, com variância de 2,2%. Na Figura 22, podemos ver a matriz de confusão do *Mover*, sendo que as linhas são as classes reais dos casos e as colunas as classes rotuladas pelo *Mover*.

Podemos observar na Figura 22 que a precisão das classes varia muito (o desvio padrão da precisão das classes é de 29%), com a classe **G** (*Generalizando um tópico*) com a pior precisão (17%) e a classe **E** (*Enunciando o tópico de pesquisa*) com a melhor (92%). Vale notar que a classe **M** (*Mostrando conhecimento da área de pesquisa*) e **I** (*Indicando uma lacuna*) têm poucos casos no conjunto de treinamento, apenas dois casos cada uma.

	M	G	I	E	R	A	Total
M	2 (28%)	0	0	0	0	0	2
G	4	17 (82%)	2	0	2	2	27
I	0	0	1 (17%)	0	0	1	2
E	0	0	0	34 (92%)	2	2	2
R	1	3	2	3	25 (66%)	8	38
A	0	1	1	0	9	16 (57%)	42
Total	7	21	6	37	38	29	138

Figura 22: Matriz de confusão do *Mover* com seu esquema original de anotação (ANTHONY; LASHKIA, 2003). Os valores entre parênteses indicam a precisão da classe

5 *Avaliação automática de qualidade de escrita*

Neste capítulo, reportamos trabalhos de avaliação de qualidade de textos, e sempre que possível focamos no gênero científico.

A avaliação da qualidade da escrita de um texto é uma tarefa subjetiva, mesmo entre humanos. Embora difícil, a avaliação da qualidade de escrita é feita em toda parte. Milhões de redações são escritas por alunos e avaliadas por professores todo ano nos ensinos Fundamental e Médio do país. E a qualidade de escrita influencia na vida das pessoas. Por ano, milhares de estudantes não ingressam em faculdades públicas devido ao mau desempenho nas provas de redação dos vestibulares.

Nessas avaliações em larga escala, em que alunos discorrem sobre o mesmo tema e o volume de dissertações torna inviável que a correção seja feita por apenas um avaliador, é necessário definir guias ou manuais com os critérios de avaliação para reduzir a subjetividade do julgamento. Há, inclusive, treinamento da equipe avaliadora para sanar dúvidas e aumentar a concordância entre os avaliadores. Os textos são avaliados por pelo menos dois juízes, e se ambos os julgamentos forem similares, é dada a nota. Se os juízes discordarem, um terceiro juiz avalia o texto, e o processo continua até que a maioria dos juízes concorde em uma nota. Com o intuito de instanciar uma aplicação real, apresentamos a seguir a rubrica utilizada no *Graduate Management Admissions Test*¹ (GMAT). Nesse teste, é pedido ao estudante que disserte sobre um determinado tema. A dissertação feita pelo estudante recebe uma nota entre 0 e 6, sendo que existe uma rubrica que define quais os requisitos mínimos que devem ser alcançados para se obter cada uma das notas.

Embora tanto a definição quanto sua apuração da qualidade de escrita sejam tarefas intrinsicamente árduas e subjetivas, a segunda tarefa é mais aceita e mais prática. E a metodologia mais utilizada para a validação de uma avaliação é a concordância entre os avaliadores.

¹<http://www.gmac.com/gmac/thegmtat/>

A score of 6 is Outstanding – A cogent, well-articulated analysis of the complexities of the issue, demonstrating mastery of the elements of effective writing, and displaying the following characteristics:

- explores ideas and develops a position on the issue with insightful reasons and/or persuasive examples is clearly well organized
- demonstrates superior control of language, including diction and syntactic variety
- demonstrates superior facility with the conventions of standard written English, but may have minor flaws

::

A score of 0 is Unscorable – A paper that is totally illegible or obviously not written on the assigned topic.

Figura 23: Trecho da rubrica utilizada no GMAT para a avaliação de textos de alunos. São mostradas os requisitos para se obter a nota máxima (6) ou mínima (0).

Assim também é medida a qualidade de escrita em sistemas automáticos. As notas dadas por esses sistemas são confrontadas com as notas atribuídas por avaliadores humanos, e é medida a concordância entre estes diferentes juízes, por meio de métricas que serão descritas no decorrer deste capítulo. Além disso, muito tempo é gasto na avaliação de *essays*. Segundo Mason & Grove-Stephenson (2002), cerca de 30% do tempo dos professores da Grã-Bretanha é gasto na correção de dissertações, e essa demanda excessiva de tempo, muitas vezes ampliada devido ao elevado número de alunos de cada professor, restringe as outras tarefas e obrigações a ele atribuídas.

Para atacar estes problemas, sistemas que avaliam automaticamente a qualidade de escrita têm sido desenvolvidos, principalmente durante os últimos quinze anos, e têm obtido altos valores de concordância com o julgamento humano, em muitos casos. Contudo, a maioria destes sistemas foca a avaliação de dissertações (*essays*). Neste trabalho focamos no gênero científico e, mais especificamente, em resumos (ou *abstracts*).

Textos do gênero científico também são largamente avaliados, por exemplo em submissões deles para conferências e revistas científicas, ou bancas de defesa de teses ou dissertações. Embora o foco neste caso seja em outros aspectos, como a relevância da pesquisa e os resultados obtidos, a forma como o pesquisador expõe sua pesquisa é de fundamental importância e impacta em sua publicação. Este fato evidencia-se mais claramente nos casos em que o pesquisador tem que escrever em uma língua que não a materna. Nesses casos, muitas vezes artigos são rejeitados pela incapacidade do escritor de “fazer-se entender” pelos

revisores.

Entretanto, a avaliação automática de textos científicos é difícil, segundo nosso ponto de vista. Um dos fatores é seu caráter inovador, que impede que gabaritos que avaliem o conteúdo sejam construídos para auxiliar na avaliação. Em dissertações com tema fixo, pode-se elaborar um gabarito com possíveis tópicos que o escritor deve abordar para lograr uma boa nota.

Sistemas de apoio à escrita científica conseguem focar apenas na adequação do texto às peculiaridades do gênero, e auxiliam o pesquisador disponha os dados de seu trabalho, isto é, a *informação factual*, de maneira clara e objetiva no texto. É desta forma que ferramentas como o SciPo, o SciPo-Farmácia e o ambiente AMADEUS se propõem a ajudar, embora com várias limitações muitas vezes desapontando seus usuários que esperam ajuda na escrita do conteúdo factual.

Sistemas de avaliação automática da qualidade de escrita são separados em dois tipos: os de **conteúdo** e os de **estilo**, de acordo com o foco da avaliação. Esta separação foi proposta por (PAGE, 1966, p. 240): “‘*Content*’ refers loosely to what the essay says, and ‘*style*’ refers to syntax and mechanics and diction and other aspects of the way it is said.”

Refraseando, sistemas que avaliam conteúdo focam na informação contida em um texto, enquanto que sistemas que avaliam estilo inspecionam a maneira como a informação está apresentada. Focamos nosso trabalho em sistemas que avaliam o estilo, pois acreditamos que a avaliação de conteúdo de um texto científico é uma tarefa nebulosa, visto que é difícil de se julgar mesmo entre avaliadores humanos.

Para diminuir a subjetividade da avaliação da qualidade de escrita de textos científicos, tem sido desenvolvida, desde 2004, uma rubrica específica para avaliar a qualidade da escrita de resumos científicos em inglês (*abstracts*) (ALUÍSIO et al., 2005; SCHUSTER et al., 2005), que será descrita na Seção 5.4.

Na Seção 5.1, apresentamos um histórico da avaliação automática de qualidade da escrita para dissertações (*essays*), principalmente. As técnicas e abordagens utilizadas são apresentadas na Seção 5.2, assim como oito sistemas que foram e estão sendo desenvolvidos (Seção 5.3). Três destes sistemas são descritos mais detalhadamente (*E-rater*, BETSY e sistema de Larkey).

Existem vários trabalhos na literatura que têm relatado a evolução dos sistemas de avaliação automática da qualidade de escrita (veja (KUKICH, 2000; VALENTI et al., 2003; MARÍN, 2004; WILLIAMS, 2001)), porém abordam sistemas diferentes e os analisam com

perspectivas diferentes. Observamos que não há um corpus padrão usado para a avaliação dos sistemas de qualidade da escrita, como hoje é comum na avaliação conjunta de outros sistemas de PLN e de Recuperação de Informação (RI) (por exemplo, na DUC² e na TREC³, respectivamente) ou na avaliação empírica de algoritmos de aprendizado de máquina em que se utiliza o repositório de bases de dados da *University of California, Irvine*⁴ (UCI), além de não haver um consenso nas medidas mais adequadas para a avaliação. Entretanto, a avaliação em larga escala começa a disponibilizar grandes corpus, provindos de testes em larga escala, como o GMAT⁵, o *Test of English as a Foreign Language*⁶ (TOEFL), o *Graduate Record Examination*⁷ (GRE) e o *General Certificate of Secondary Education*⁸ (GCSE). Muitos dos sistemas que descreveremos nesta seção surgiram em análises das dissertações desses testes e muitos foram avaliados com as pontuações manuais dadas a essas dissertações e as pontuações obtidas pelos sistemas automáticos.

5.1 Histórico

A pesquisa de sistemas que avaliam automaticamente dissertações (*essays*) produzidas por alunos iniciou-se em meados da década de 1960, com Ellis Page. Page desenvolveu um sistema de avaliação (ou pontuação) automática da qualidade de escrita, o *Project Essay Grader* (**PEG**) (PAGE, 1966, 1994). A pesquisa ficou estagnada até a década de noventa, quando o avanço de outras áreas, principalmente do PLN e da Recuperação da Informação (RI), possibilitou abordagens mais consistentes e profundas para avaliar automaticamente a qualidade de textos. A Figura 24 mostra a evolução e o surgimento de sistemas de avaliação automática da qualidade de escrita no decorrer dos anos.

No início da década de 1990, entra em cena a organização ETS⁹ (*Education Testing Service*), que se consolidaria nos anos seguintes no desenvolvimento de sistemas automáticos de avaliação de escrita. Em 1994, a ETS lançou, utilizando técnicas de PLN, o sistema **ETS I**. Page retoma o desenvolvimento do PEG, que em 1997 tornou-se comercial. Logo, outros novos surgiram: o **Intelligent Essay Assessor (IEA)** (HEARST, 2000), primeiro sistema baseado em Semântica Latente ou LSA (*Latent Semantic Analysis*) (LANDAUER et

²<http://duc.nist.gov/>

³<http://trec.nist.gov/>

⁴<http://www.ics.uci.edu/~mlearn/MLRepository.html>

⁵<http://www.gmac.com/gmac/thegmat/>

⁶<http://www.ets.org/toefl/>

⁷<http://www.ets.org/gre/>

⁸<http://www.gcse.com/>

⁹<http://www.ets.org/>

Figura 24: Evolução dos sistemas no tempo (MARÍN, 2004).

al., 1998); o **E-rater** (BURSTEIN et al., 1998), desenvolvido pela ETS, combinando técnicas de PLN e estatística; e o **IntelliMetric**¹⁰, que utiliza conceitos de Inteligência Artificial. Em 1998, a ETS adaptou o **E-rater**, gerando o **C-rater**. Com um investimento de mais de um milhão de dólares, foi criado um projeto na ETS denominado **Criterion** (BURSTEIN et al., 2003), que resultou na interface web *Criterion 1.0*, implementada sobre o **E-rater**. O último release foi o *Criterion v7.1*¹¹, de janeiro de 2007.

A partir do final dos anos noventa, muitos outros sistemas foram desenvolvidos: **SEAR** (*Schema Extract Analyze and Report*) (CHRISTIE, 1999), baseado em casamentos de padrão (*pattern-matching*); **Automark** (MITCHELL et al., 2002), que usa técnicas de PLN; **IEMS** (*Intelligent Essay Marking Systems*) (MING et al., 2000) baseada na técnica Indextron; **Apex Assessor** (DESSUS et al., 2000), que utiliza a LSA, assim com o IEA; o **Automated Text Marker** (ATM) (CALLEAR et al., 2001), que busca conceitos; o **sistema de Larkey**, que combina classificadores Bayesianos com outras técnicas estatísticas; **BETSY** (*Bayesian Essay Test Scoring sYstem*) (RUDNER; LIANG, 2002), baseado em análise estatística e classificação de textos Bayesiana e o **PS-ME** (*Paperless School free text Marking Engine*) (MASON; GROVE-STEPHENSON, 2002), baseado em PLN.

¹⁰<http://www.vantagelearning.com/intellimetric/>

¹¹<http://criterion2.ets.org/cwe/>

Em 2003, dois novos sistemas surgiram: o *Auto-marking* (SUKKARIEH et al., 2003), baseado em PLN; e o *CarmelTC* (ROSÉ et al., 2003), baseado em técnicas de aprendizado de máquina e classificação bayesiana.

5.2 Técnicas utilizadas na avaliação automática da qualidade de escrita

Nesta seção, descrevemos as técnicas mais estudadas em sistemas automáticos de avaliação da qualidade de escrita. Alguns sistemas utilizam mais de uma técnica. Citamos os tipos mais comuns de modelagem dos textos usados por cada uma das técnicas. Descrevemos também o método de redes complexas (ANTIQUERA, 2007) que, embora não seja parte de nenhum sistema, tem sido estudado e mostrado que algumas das métricas da rede têm alta correlação com a qualidade dos textos avaliada manualmente.

5.2.1 Técnicas de PLN

Ferramentas de PLN são muito utilizadas nas etapas de pré-processamento e modelagem dos textos. São utilizadas na etapa de pré-processamento de praticamente todos os sistemas descritos. Dentre os tipos de ferramentas aplicadas, as mais comuns são os sentenciadores, os tokenisadores, a remoção de *stopwords*¹², a lematização e o *stemming*¹³.

Alguns sistemas utilizam ferramentas de PLN para obter estruturas lingüísticas mais complexas do texto, que podem ser utilizadas como atributo, por exemplo. O *E-rater*, descrito na Seção 5.3.1, é o melhor exemplo do uso deste tipo de ferramenta.

Sistemas que usam PLN são dependentes de língua, e sua migração para uma outra língua é demasiadamente custosa. Embora os resultados da avaliação sejam mais aceitos, a construção de sistemas que envolvem muito PLN requer grandes equipes e investimentos.

5.2.2 Técnicas de Recuperação de Informação

As métricas de Recuperação de Informação (RI) são utilizadas principalmente para estimar a similaridade e para fornecer ranqueamento. Diferentes medidas de RI são utilizadas para

¹²*Stopwords* são palavras muito freqüentes de uma língua, como artigos, preposições, etc.

¹³Lematização e *stemming* são processos semelhantes, aplicados a palavras para remover marcas morfológicas, como plural e flexão verbal. Ambos reduzem palavras a seus respectivos *stems*. A diferença entre eles é que o lematizador usa informações morfosintáticas, para distinguir entre ‘casa, verbo’ de ‘casa, substantivo’, por exemplo. No *stemming* seriam reduzidas a um mesmo *stem*

o ranqueamento de atributos em modelagens *bag of words*, como métrica para selecionar os atributos mais relevantes. Na modelagem *bag of words*, cada texto é representado por uma lista de frequência (ou apenas ocorrência) das palavras de seu próprio texto. Uma das medidas mais utilizadas é a $TF*IDF$, embora outras medidas como entropia e *information gain* também sejam utilizadas (para uma revisão mais ampla de métricas de RI, veja (MANNING; SCHÜTZE, 1999; BAEZA-YATES; RIBEIRO-NETO, 1999)).

5.2.3 Técnicas Categorização Textual

Categorização Textual (SEBASTIANI, 2002) é a tarefa de separar documentos ou textos em determinados grupos ou categorias ou pré-estabelecidos. Para avaliação da escrita, as categorias podem ser as notas dadas ao texto ou uma classificação binária, como “bons” e “ruins”. O modelo de representação mais comum dos documentos neste tipo de abordagem é o *bag of words*. Os atributos utilizados na classificação são as palavras, sendo que vários métodos são usados para diminuir o número de atributos. Entre os mais usados estão duas etapas de pré-processamento comuns: remoção de *stopwords* e lematização.

Sobre os valores dos atributos extraídos dos textos são aplicados algoritmos de aprendizado supervisionado. Os algoritmos mais utilizados em categorização textual são, principalmente, o *Naive Bayes* e o SVM. Estes dois algoritmos têm obtido bom resultados nesses tipos de tarefa (JOACHIMS, 1998; GONÇALVES, 2002).

5.2.4 *Vector Space Model*

Vector Space Model (VSM) (SALTON et al., 1975) modela cada texto como um vetor, no qual as dimensões correspondem aos atributos, que são um subconjunto das palavras presentes nos textos do treinamento. Os valores dos atributos de um texto são as frequências das palavras. A comparação dos textos é feita pelo co-seno dos vetores associados a cada texto, e quanto menor o valor do co-seno, mais similares são os textos. Esta técnica permite que o treinamento seja feito com apenas uma classe como, por exemplo, apenas textos considerados bons. Quando um novo texto é classificado, primeiramente ele é transformado para a representação vetorial e então é comparado os textos do treinamento, e podemos estabelecer um valor máximo do co-seno. Os textos rotulados como “bom” precisariam de um valor de co-seno, com quaisquer outros textos do treinamento, menor que o valor máximo.

5.2.5 Análise Semântica Latente

Latent Semantic Analysis (LSA) (LANDAUER et al., 1998) é uma medida estatística complexa que foi aplicada inicialmente para indexação de documentos e recuperação de informação. A LSA representa o texto em uma matriz, sendo que as linhas correspondem a termos e as colunas correspondem aos documentos. A LSA transforma essa matriz em uma relação de termos com *conceitos*, e uma relação entre esses conceitos e documentos. Dessa forma, documentos e termos estão indiretamente relacionados. LSA pode ser descrita resumidamente nos passos abaixo (MARÍN, 2004):

- 1.
2. Fase de treinamento: cálculo dos pesos para os vetores que representam os textos de referência.
3. Fase de testes: cada texto deve estar no formato LSA. Para tanto, algumas transformações devem ser feitas:
 - (a) Representação matricial: exclui-se *stopwords* e as palavras restantes são lematizadas e armazenadas na matriz. Essa matriz pode ser vista como a representação VSM de contextos (sentenças, parágrafos e documentos).
 - (b) Acerto dos pesos da matriz: é calculada a relevância de cada termo em cada trecho. A frequência de cada termo em todos os possíveis contextos é medida para identificar se ela se sobressai em algum desses contextos.
 - (c) *Singular Value Decomposition* (SVD): A matriz original é decomposta no produto de três matrizes ortogonais. Uma delas é diagonal, e seus valores são os valores singulares da matriz original.
 - (d) Transformação de redução de dimensionalidade: para extrair as relações relevantes entre termos e contextos, é necessário reduzir o ranking da matriz diagonal. Nesta transformação é que as relações escondidas são detectadas.
 - (e) A reconstrução: Essa nova matriz diagonal é multiplicada pelas duas outras matrizes obtidas no SVD e o resultado é uma versão com pesos da matriz original, que é a representação LSA do texto.
4. Fase de aplicação: A representação LSA obtida no passo anterior é confrontada com as representações LSA dos modelos de referência, e são ranqueados os textos do treinamento mais similares ao texto submetido.

5.2.6 Redes Complexas

As redes complexas são basicamente grafos, com a ressalva de que apresentam uma diversidade de técnicas e modelos que auxiliam no entendimento e previsão do comportamento dos sistemas. Três conceitos merecem destaque no estado da arte em redes complexas (ALBERT; BARABASI, 2002): as redes *small-world*, o coeficiente de aglomeração (*clustering coefficient*) e as redes livres de escala (*scale-free*).

O conceito *small-world* refere-se ao fato de que, mesmo enormes, a maioria das redes apresenta um caminho relativamente curto entre quaisquer dois nós. E uma rede é dita livre de escala se a probabilidade $P(k)$ de um vértice possuir k arestas obedece a uma distribuição $P(k) \sim k^{-\gamma}$. O termo “livre de escala” refere-se à essa propriedade, uma vez que essa distribuição é independente do número total de vértices (escala) do grafo.

O processo de modelagem de um texto em uma rede complexa pode ser analisado mais claramente em (ANTIQUERA, 2007). Simplificadamente, as palavras do texto são os nós da rede, e as arestas existentes na rede refletem a adjacência dessas palavras no texto. Os pesos das arestas são as frequências da ocorrência dos termos consecutivos, no sentido da orientação da aresta, no corpus.

Apenas recentemente, redes complexas têm sido aplicadas a diversos problemas de PLN, como extração terminológica, sumarização (ANTIQUERA, 2007) e também para avaliar a qualidade de textos (ANTIQUERA et al., 2007, 2005). Estes estudos indicam que algumas métricas de redes complexas estão correlacionadas com a qualidade de um texto, como é analisado por (ANTIQUERA, 2007, p. 30-31):

“Os autores observaram que, conforme o grau e o coeficiente de aglomeração aumentam, a qualidade dos textos tende a diminuir. Essa constatação indica que, quando o número de conexões entre as palavras de um texto é excessivo, sua qualidade tende a cair. No caso da dinâmica do número de componentes, pôde-se perceber que, quanto mais cedo novos conceitos são apresentados no texto, pior o texto fica. A dinâmica do número de componentes quantifica a velocidade com que novas palavras são utilizadas em um texto.”

São duas as medidas que apresentaram correlação com a qualidade de escrita. A dinâmica do número de componentes quantifica a velocidade com que novas palavras são utilizadas em um texto. O coeficiente de aglomeração quantifica a conectividade de um

vértice com seus vizinhos. Segundo Antiqueira (2007), em vértices com alto coeficiente de aglomeração, significa que seus vizinhos estão bem conectados entre si.

5.3 Descrição dos sistemas

Nesta seção, descrevemos alguns dos sistemas citados na Seção 5.1 mais detalhadamente, indicando as técnicas/abordagens utilizadas e, em alguns casos, em quais aplicações foram utilizados, além de peculiaridades dos sistemas. Damos enfoque à três sistemas, todos de avaliação de estilo: ao *E-Rater* (Seção 5.3.1), por ser o sistema mais utilizado e ter uso prático em avaliações diversas; o BETSY (Seção 5.3.2), por ser o único sistema livre, que pode ser testado; e o sistema de Larkey (Seção 5.3.3), que foi pioneiro em modelar o problema de avaliação como um problema de Categorização Textual.

Alguns outros sistemas são citados no final desta seção, juntamente com informações sobre o desenvolvedor e as técnicas que empregam.

5.3.1 *E-rater*

O *Electronic Essay Rater*¹⁴ ou ***E-rater*** (BURSTEIN et al., 1998, 2001) é a continuação do ETS I da ETS. Esse sistema usa um método híbrido de identificação dos atributos a serem utilizados, incluindo 3 análises: análise da estrutura sintática, análise da estrutura retórica e análise do tópico a ser percorrido. Uma das principais qualidades desse sistema é ele se basear nos atributos especificados em rubricas¹⁵ também utilizadas por humanos na correção de textos em exames aplicados pelo ETS como o GMAT. Os atributos utilizados pelo ***E-rater*** ou foram derivadas estatisticamente ou extraídas por meio de técnicas de PLN. A arquitetura do sistema, que pode ser vista na Figura 25, é detalhada abaixo.

1. **Atributos sintáticos** Nesta avaliação, estima-se a variedade sintática do texto. Todas as sentenças do texto são analisadas sintaticamente. Existe um programa no *E-rater* para identificar cláusulas complementares, subordinativas, infinitivas e relativas, a presença de verbos auxiliares modais (*would*, *could*, *should*, *might* e *may*). As razões dos tipos de estruturas sintáticas por texto e por sentença também são utilizadas como medidas da variação sintática.

¹⁴<http://www.ets.org/research/erater.html>

¹⁵ *Analytical Writing Assessment (AWA) Scoring Guide*:
<http://www.gmac.com/gmac/TheGMAT/GMATScores/UsingScorestoAssessIndividuals/UnderstandingandUsingtheAnalyticalWritingAssessment>

Figura 25: Arquitetura do *E-rater*.

2. **Análise da Estrutura Retórica** As rubricas de pontuação indicam a nota a ser dada se o aluno apresentar um texto bem desenvolvido. A abordagem utilizada para o reconhecimento da estrutura e das relações retóricas é a identificação de expressões-padrão (*cue words*). A abordagem de identificar as expressões-padrão, sustentada pela literatura da área de análise do discurso (*apud* (BURSTEIN et al., 1998)), supõe que essas expressões marcam a presença de um novo argumento. O *E-rater* usa um programa denominado APA (*automated argument partitioning and annotation program*) para construir uma versão do texto segmentada por argumentos, em que o texto é separado em unidades argumentativas, que são classificadas como, por exemplo, de “início de um argumento” ou “desenvolvendo o argumento”. O APA utiliza regras para a anotação de argumentos e a segmentação é baseada na estrutura sintática e na distribuição, por parágrafo, das expressões-padrão, frases e estruturas para identificar a estrutura retórica.
3. **Análise do Tópico** Nessa análise, bons textos usam um vocabulário mais preciso e especializado, além de textos do mesmo nível serem semelhantes quanto ao vocabulário usado. O *E-rater* compara as palavras de um texto a ser testado com as palavras encontradas nos textos de treinamento, agrupados em seis diferentes grupos pela nota. São dois programas utilizados para computar a similaridade de conteúdo, baseados em duas abordagens distintas: o *EssayContent*, baseado na frequência das palavras e o *ArgContent*, que utiliza conceitos de RI para calcular o “peso” de cada palavra. A técnica VSM (descrita na Seção 5.2.4) é utilizada para o cálculo da similaridade em ambos os programas.

As análises feitas pelo *E-rater* resultaram em 57 atributos que podem ser usados (BURSTEIN et al., 1998), que são selecionadas com o auxílio do método de regressão linear. O sistema precisa ser treinado, com pelo menos 200 textos do tópico a ser avaliado, e retorna

uma pontuação entre 0 (ilegível ou fora do tópico proposto) e 6 (ótimo). O *E-rater* foi utilizado em mais de 750.000 dissertações do GMAT, com cerca de 90% de correlação com o outro anotador, que foi o mesmo valor apurado entre dois anotadores humanos. Além disso, o ***E-rater*** é o módulo funcional usado na aplicação *Criterion*¹⁶ da ETS que avalia habilidades de escrita de estudantes, fornecendo a pontuação (veja no Apêndice A, a Figura 35 em que o *Criterion* avalia a redação submetida e o *feedback* para o estudante pode ser visto na Figura 36).

5.3.2 BETSY

BETSY¹⁷ (RUDNER; LIANG, 2002) é um sistema que classifica o texto com base em textos previamente classificados, que são utilizados na fase de treinamento do sistema. Segundo Rudner & Liang (2002), o diferencial deste sistema consiste em determinar a categoria do texto (numa escala de 4 notas, como, por exemplo, abrangente, essencial, parcial e insatisfatório) por meio do uso de um vasto conjunto de atributos, tanto para determinar características de estilo quanto de conteúdo. O BETSY é fundamentado em redes Bayesianas. Primeiramente, são removidas *stopwords* do texto, e as palavras restantes são lematizadas. Para a extração dos atributos, dois modelos podem ser usados: o Modelo Multinomial (MM) e modelo de Bernoulli (MB). O MB é mais rápido, pois apenas busca a presença de certos atributos, enquanto que, no MM, todos os atributos do vocabulário precisam ser examinados. Uma comparação entre os modelos é feita em (RUDNER; LIANG, 2002) e eles sugerem que o MM com um vasto vocabulário é mais precisa que o MB, embora os autores ressaltem que em classificações baseadas na presença ou ausência de conceitos-chave talvez a afirmação não se sustente. Para selecionar os melhores atributos, foi utilizada uma medida da teoria da informação denominada entropia (MANNING; SCHÜTZE, 1999). Foi utilizado um corpus de 462 textos para treinamento e 80 textos para os testes. A taxa de acerto alcançada foi de 80%. O BETSY é o único dos sistemas estudados que não é comercial e está disponível, podendo ser usado livremente, desde que mantidos os créditos aos criadores. Uma das desvantagens é que o BETSY é que seu treinamento é lento, e é restrito ao ambiente Windows (foi escrito em Power Basic).

¹⁶<http://criterion.ets.org>

¹⁷<http://edres.org/betsy/>

5.3.3 Sistema de Larkey (*Larkey's System*)

O sistema de Larkey (LARKEY, 1998) trata a qualidade de escrita como uma tarefa de Categorização Textual. Nele, as categorias a serem identificadas são as notas que se pretende dar a um texto. Foi desenvolvido para classificar textos como “*bons*” ou “*ruins*”, e é um dos únicos sistemas em que os textos *ruins* são incorporados ao treinamento. Para a classificação, o sistema de Larkey testou três distintas abordagens na indução de classificadores, além de algumas combinações delas.

1. **Classificadores Bayesianos:** são utilizados para estimar a probabilidade de um documento ser de uma determinada categoria, dada à presença de certas palavras no documento. Primeiramente, *stopwords* são removidas, e as palavras restantes são lematizadas, e os documentos são representados na forma *bag of words*. Os atributos foram ranqueados segundo o *Expected Mutual Information* (EMIM), calculado para cada um dos atributos. Os melhores atributos são selecionados, sendo que o número de atributos é calibrado no cópulus de treinamento. Os classificadores bayesianos são treinados usando o modelo binário de Lewis (LEWIS, 1998), não é considerada a frequência dos termos, apenas se este está presente ou não no texto. No sistema de Larkey, são induzidos vários classificadores binários: para textos pontuados numa escala de quatro notas (1, 2, 3 e 4), por exemplo, foram induzidos três classificadores: um para distinguir textos com nota 1 dos textos com notas 2, 3 e 4; um para distinguir textos com notas 1 ou 2 dos textos com nota 3 ou 4; e um para distinguir textos com nota 4 de textos com notas 1, 2 e 3.
2. **Classificação por “*k* vizinhos mais próximos” (kNN):** na classificação por kNN, são procurados os *k* textos no conjunto de treino que são mais similares ao texto submetido à comparação. A similaridade entre textos foi medida pela medida *TF*IDF*, com o auxílio do sistema de RI *Inquery* (CALLAN et al., 1995). O novo texto é enviado ao *Inquery* como sendo uma consulta contra os documentos do conjunto de treino. O ranqueamento retornado foi utilizado como métrica de similaridade. A pontuação do texto submetido é a média ponderada dos valores das notas por similaridade dos *k* textos no topo do ranqueamento. O valor de *k* é calibrado no próprio conjunto de treino.
3. **Text-Complexity features:** são atributos simples que extraem características superficiais do texto, como o número de palavras e número de palavras com pelo menos 6 letras. Os 11 atributos utilizados são descritos na Figura 26.

Atributo	Descrição
L1	Número de caracteres no documento
L2	Número de palavra no documento
L3	Número de palavras distintas no documento
L4	A raiz quarta do número de palavras no documento
L5	Número de sentenças no documento
L6	Tamanho médio das palavras ($L1/L2$)
L7	Tamanho médio das sentenças, em número de palavras ($L2/L5$)
L8	Número de palavras com mais de 5 caracteres
L9	Número de palavras com mais de 6 caracteres
L10	Número de palavras com mais de 7 caracteres
L11	Número de palavras com mais de 8 caracteres

Figura 26: Os 11 atributos superficiais (*Text-complexity features*) utilizados no sistema de Larkey (LARKEY, 1998)

Além dos classificadores sozinhos, foram testadas combinações desses classificadores, sobre três combinações de variáveis:

- (1) as onze *text-complexity features*;
- (2) apenas os classificadores Bayesianos;
- (3) todas as variáveis: as onze *text-complexity features*, as notas dadas pelo classificador kNN, e as notas dadas pelos classificadores Bayesianos.

Foram realizados testes com corpus de gêneros diferentes: textos das áreas de Ciências Sociais, Física, Direito e dois conjuntos de redações. Todos os conjuntos utilizam uma escala de 4 possíveis notas, com exceção dos textos de Direito, com uma escala de 7 notas. Três medidas de comparação foram utilizadas: taxa de acerto, a proporção de casos cuja nota fosse até um ponto de diferença da classificação real (*Adjacent*) e o coeficiente de correlação de Pearson¹⁸. A taxa de acerto ultrapassou 60% e a medida *Adjacent* alcançou 100% nos textos de Ciências Sociais. Nos textos de Direito, a correlação de Pearson foi de 77%, e nas redações chegou a 88%. Entretanto, nesses mesmos redações a taxa de acerto foi de, no máximo, 55%, o mesmo relatado entre dois anotadores humanos.

5.3.3.1 Outros sistemas

Vários trabalhos apresentam um histórico e breve descrição de sistemas de avaliação automática da qualidade de escrita (KUKICH, 2000; MARÍN, 2004; VALENTI et al., 2003;

¹⁸O coeficiente de Pearson estima a correlação linear entre duas variáveis aleatórias. Varia entre -1 e 1, sendo 1 é a correlação perfeita e positiva; -1 é a correlação perfeita, mas inversa; e 0 indica que não há correlação linear entre as duas variáveis.

WILLIAMS, 2001). Aqui descrevemos resumidamente outros sistemas comerciais encontrados na literatura.

1. ***Project Essay Grader (PEG)*** (PAGE, 1994): Utilizando dissertações avaliadas anteriormente por professores, Page levantou características que podiam ser automaticamente medidas e, por meio de métodos numéricos, determinou pesos que mediam a influência destas na qualidade do texto para cada medida. No sistema, as características extraídas automaticamente restringiam-se às superficiais ao texto, como tamanho do texto em palavras, número de vírgulas, preposições, entre outras. Page denominou essas características de *proxes*, como medidas indiretas que intrinsecamente mediam a qualidade de escrita uma redação. O PEG alcançou 78% de correlação de regressão múltipla, muito próxima dos 85% obtidos entre dois ou mais professores. Apesar dos ótimos resultados acima, o PEG não foi muito aceito, pois as qualidades importantes do texto, tais como conteúdo, organização e estilo, não eram consideradas. Os alunos podiam artificialmente simular um bom texto manipulando as *proxes*, inserindo mais palavras ou preposições, por exemplo.
2. ***Intelligent Essay Assessor (IEA)*** (HEARST, 2000): é baseado na técnica LSA. Começou como um projeto acadêmico, mas os criadores acabaram fundando uma empresa chamada *Knowledge Analysis Technology*, e estão patenteando o sistema. O diferencial do IEA é que ele acessa o conhecimento contido no texto, mais que estilo, sintaxe e estruturação dos argumentos.
3. ***Intelligent Essay Marking Systems (IEMS)*** (MING et al., 2000): é baseado na técnica Indextron (*Pattern Indexing Neural Network*), desenvolvida na Politécnica de Ngee ANN. Indextron é definido como um algoritmo de clusterização. O uso do Indextron com base para o desenvolvimento de redes neurais é uma tentativa de superar o treinamento lento e não-incremental característico de redes neurais artificiais tradicionais.
4. ***Schema Extract Analyse and Report (SEAR)*** (CHRISTIE, 1999): é um sistema desenvolvido em 1999 na Universidade Robert Gordon de Aberdeen, Inglaterra, e trabalha com algoritmos diferentes para levantar os atributos de contexto e de estilo. Para conteúdo, o sistema precisa de esquemas de referência sejam criados e passados ao sistema. Para avaliar o estilo, primeiro são determinadas algumas métricas-candidatas; segundo, deve ser dado um conjunto de treinamento marcado manualmente; então um

processo de calibragem é iniciado, até que a concordância entre a pontuação manual e a automática seja aceitável.

5. **PS-ME** (MASON; GROVE-STEPHENSON, 2002): o *Paperless School Marking Engine*¹⁹ foi desenvolvido por MASON; GROVE-STEPHENSON na Universidade de Birmingham, Inglaterra, e tornou-se comercial. Este sistema utiliza técnicas de PLN para avaliar o texto por meio do conhecimento, entendimento e avaliação. O PS-ME deve ser treinado inicialmente, e o conjunto de treinamento pode ter não somente textos-referência, mas também textos ruins (maus exemplos). Além disso, O PS-ME retorna não só a pontuação do texto, mas também *feedback* dos erros cometidos.

5.3.4 Categorização dos sistemas de avaliação da qualidade de escrita

Como já antecipamos no início deste capítulo, sistemas de avaliação automáticos de qualidade de escrita podem ser divididos de acordo com o tipo de informação que este tentam mensurar: os que avaliam **conteúdo** e os que avaliam **estilo**. Os que avaliam conteúdo focam no significado do texto como, por exemplo, avaliar se uma redação explora o tema proposto. Sistemas que avaliam estilo julgam a organização, a sintaxe, a mecânica e o estilo de escrita. Essas características são muito difíceis de se obter pelo processamento computacional, e quase sempre acabam sendo medidas indiretamente. Page afirma que não é possível isolar e medir as variáveis intrínsecas que são o objetivo de uma avaliação do estilo, como fluência, pontuação e gramática, mas que estes quesitos podem ser medidos por meio de outras medidas simples, cuja correlação com o objetivo de avaliação for alta. Por exemplo, a fluência tem alta correlação com o número de palavras de um texto, e então a fluência é medida por meio do número de palavras. Entretanto, atualmente sistemas têm obtido medida de maneira mais profunda, como o *E-Rater*, que mede por exemplo a variação de estruturas sintáticas para medir fluência. Atualmente, os sistemas mais modernos convergem para sistemas de avaliação automática cuja metodologia seja parecida com a avaliação humana.

A metodologia usada pela ferramenta para medir estilo e conteúdo é outra característica que pode ser utilizada para dividir sistemas. Segundo Williams (2001), os sistemas podem ser classificados como *Rating Simulation*, se a avaliação é feita por meio de medidas correlatas ou superficiais; ou *Master Analysis*, caso busquem mensurar as dimensões diretamente, por variáveis que sejam condizentes em uma avaliação conduzida manualmente.

¹⁹<http://www.paperless-school.com>

Podemos ver na Tabela 8 exemplos de sistemas que avaliam conteúdo ou estilo, também separados pela maneira como avaliam-nas:

Tabela 8: Sistemas divididos por: conteúdo ou estilo e *Rating Simulation* ou *Master Analysis* (VALENTI et al., 2003).

	<i>Rating Simulation</i>	<i>Master Analysis</i>
Conteúdo	IEA, BETSY, IEMS, SEAR	ETS I, E-Rater, C-Rater, Automark, PS-ME
Estilo	PEG, BETSY, IEMS, SEAR	<i>E-Rater</i> , Automark, PS-ME

Mostramos resultados de avaliação para alguns dos sistemas. Uma comparação mais ampla é feita em (MARÍN, 2004). A Tabela 9 mostra os sistemas que descrevemos, com suas respectivas técnicas e resultados.

Tabela 9: Indicação das técnicas utilizadas pelos sistemas, resultados e onde foram aplicados (MARÍN, 2004). A sigla **corr** significa correlação; **conc**, concordância; **acc**, taxa de acerto. Quando os autores apresentam diversos valores, o menor foi o considerado

Sistema	Técnica	Resultados	Aplicações
BETSY	Estatístico, Bayesiano	ac:0,8	Textos quaisquer
Sistema de Larkey	Categorização Textual	ac:0,8	Ciências sociais e dissertações gerais
<i>E-Rater</i>	PLN, estatístico	conc:0,97	GMAT
PEG	Estatístico	corr:0,87	Redações de tema aberto
IEA	LSA	conc:0,85	Redações de psicologia e de militares
IEMS	Indextron	corr:0,8	relatório de projeto de estudantes de engenharia
SEAR	PLN	corr:0,3	Dissertações de história
PS-ME	PLN	—	—

5.4 Uma rubrica específica do gênero científico para avaliação automática da qualidade de escrita

A necessidade de definir quais atributos são importantes na avaliação de textos científico reuniu pesquisadores em torno do desenvolvimento de um guia para avaliação da qualidade de um resumo escrito em inglês²⁰. Esta rubrica vem sendo desenvolvida desde de 2004 por pesquisadores do NILC e da Northern Essex Community College, EUA (ALUÍSIO et al., 2005; SCHUSTER et al., 2005). Esses pesquisadores investigaram os erros gramaticais

²⁰Rubrica é um conjunto de notas, apontamentos; indicação geral do assunto ou categorias de algo. Em ingles, o termo *rubric* significa “a set of instructions” (um conjunto de instruções)

mais comuns cometidos por estudantes brasileiros ao escreverem em inglês. No todo, foram analisados 114 resumos científicos escritos por estudantes das áreas de Farmácia, Bioquímica, Biologia/Genética, Física e Ciências da Computação em cursos de escrita técnica. A descrição dos 23 tipos de erros detectados gramaticais, assim como a distribuição dos erros no 114 resumos podem ser vistas no Apêndice B. Além de erros gramaticais, a rubrica avalia outras dimensões da qualidade como organização retórica, coerência e coesão, estilo e informatividade.

Os objetivos dessa rubrica são:

1. Ser reproduzível, isto é, que cada tarefa determinada para medir cada um dos aspectos de qualidade apurados seja reproduzível, reduzindo a subjetividade da avaliação. Este fator é fundamental para a automatização da tarefa.
2. Ser incorporada em sistemas computacionais, para seleção de corpus e crítica com *feedback*. Sua automatização pode ser incorporada tanto a sistemas usados para selecionar textos de um corpus para fazer parte de ferramentas de suporte à escrita quanto em sistemas de suporte à escrita, para possibilitar a promoção de *feedback* consistente ao texto do aluno.

Atualmente, essa rubrica é composta de sete dimensões, sendo que cada dimensão aborda diferentes aspectos de qualidade a serem medidos, principalmente aqueles que tendem a ser críticos para escritores não-nativos da língua inglesa. Cada uma das dimensões pode assumir dois valores: alto, se os aspectos de qualidade daquela dimensão foram cumpridos satisfatoriamente; ou baixo, caso contrário. As dimensões D1 e D2 são aplicadas ao resumo como um todo, enquanto que as outras cinco dimensões são aplicadas à cada sentença. As dimensões que avaliam cada uma das sentenças nem sempre são aplicadas a todas as sentenças do resumo, pois algumas dimensões focam sentenças de certas componentes da estrutura esquemática específicas. Neste último caso, estas sentenças recebem o valor *NA* (não se aplica).

D1 – *Caracterização, organização e desenvolvimento* . Esta dimensão trata da estrutura do resumo, focando tanto a presença de componentes essenciais quanto a sua ordem no texto.

- **Alto**

- Componentes principais presentes e são apresentados em ordem: Propósito, Metodologia (se houver), Resultados e Conclusão;

- Se houver uma Lacuna, deve ser seguida pelo Propósito;
- Se existir Contexto e Lacuna, devem aparecer na ordem descrita (pode haver ciclos de Contexto e Lacuna)

- **Baixo:** Caso contrário.

D2 – *Balanceamento entre os componentes* . Os resumos em geral não devem ultrapassar um limite de 200 a 300 palavras, o que impõe restrições a certos componentes estruturais, como Contexto.

- **Alto**

- Propósito existe e foi escrito em apenas uma sentença
- Conclusão existe e foi escrita em apenas uma sentença
- Se existir *Contexto*, esta componente não deve ultrapassar 30% das palavras do resumo²¹

- **Baixo:** Caso contrário.

D3 – *Coerência entre os componentes* . Os componentes de um resumo devem ser relacionados entre si, de forma a contribuir com a coerência do texto. A classificação é dada aos pares de sentenças relacionadas.

- **Alto**

- Se houver Lacuna, o Propósito de estar relacionado com ela (relação de *fulfilment*)
- Se o Resultado estiver relacionado com o Propósito (relação de *accomplishment*)
- Se a Conclusão estiver relacionada com o Resultado (relação de *generalization*)

- **NA:** Outras sentenças

- **Baixo:** Determinado para as componentes iniciais (destacadas acima) caso não haja as relações citadas entre eles.

D4 – *Marcadores de coesão* . As sentenças de cada componente devem ser coesas. A coesão pode ser alcançada por meio do uso de marcadores discursivos, referências pronominais e reintrodução de nomes. São avaliadas todas as sentenças pertencentes às componentes esquemáticas representadas por no mínimo duas sentenças.

²¹Este valor de 30% foi estimado no corpus de resumos da ferramenta SciPo-Farmácia

- **Alto**

- Se cada sentença é relacionada com pelo menos uma outra sentença da mesma categoria esquemática.

- **NA:** Se a categoria esquemática é representada por apenas uma sentença. Note: Ciclos de Contexto e Lacuna são considerados como um único componente nessa dimensão.

- **Baixo:** Caso contrário.

D5 – *Erros técnicos* . São representados por seis tipos de erro técnicos ou gramaticais:

1. Uso incorreto de uma palavra pra expressar um significado pretendido (**WU**);
2. Ausência de um artigo necessário (**ART-**);
3. Pontuação (**P**);
4. Ortografia (**SP**);
5. Uso incorreto de itens lexicais e colocações recorrentes (**WUCol**);
6. Presença desnecessária de um artigo (**ART+**).

- **Alto:** Sem erros.

- **Baixo:** Se houver pelo menos um erro.

D6 – *Estilo* . Espera-se que um texto científico não tenha um estilo coloquial e empregue expressões características do gênero.

- **Alto**

- Ausência de termos como *I, my, lot, for sure, I think, kind of, you know, I mean, I think, I assume, sort of*.

- **Baixo:** Caso contrário.

D7 – *Informação factual* . Embora alguns autores prefiram resumos indicativos, espera-se que os resumos sejam informativos, ou seja, tragam informações relevantes sobre o trabalho em questão.

- **Alto**

- Se as sentenças de Resultado e Conclusão são informativas

- **Baixo:** Caso contrário.

- **NA:** Sentenças de outras componentes esquemáticas.

Para verificar o grau de concordância entre diferentes juízes avaliando os mesmos resumos, foi realizado um experimento de reprodutibilidade com base na medida *Kappa*. Na Tabela 10, são mostradas as sete dimensões e seus respectivos valores de *Kappa* obtidos entre três anotadores humanos. Duas dimensões obtiveram valores baixos de *Kappa*. A dimensão D6 será redefinida com a inclusão de uma lista maior de termos. A dimensão D5, que recebera um valor próximo de 0, foi reformulada, embora ainda não se tenha repetido os testes para se mensurar o *Kappa* atual desta dimensão.

É importante ressaltarmos que apenas as dimensões D5 e D6 não utilizam a estrutura esquemática do resumo, como pode observado na última coluna da Tabela 10. As outras cinco dimensões não podem ser aplicadas sem que a estrutura esquemática do resumo esteja acessível, isto é, cada sentença do resumo deve estar marcada com uma categoria esquemática.

Tabela 10: Dimensões da rubrica. O *Kappa* relatado foi obtido entre anotadores humanos. A dimensão D5 foi reformulada

Dimensão	Descrição	<i>Kappa</i>	Depende da EE
D1	Caracterização, organização e desenvolvimento	1	Sim
D2	Balanceamento entre componentes	1	Sim
D3	Coerência entre componentes	0.769	Sim
D4	Marcadores de coesão	0.70	Sim
D5	Erros técnicos	–	Não
D6	Estilo	-0.023	Não
D7	Grau de informatividade	0.864	Sim

As dimensões em que os juízes obtiveram maior grau de concordância foram a D1, D2 e D7 ($K=1$, $K=1$ e $K=0,864$, respectivamente), sendo consideradas reprodutíveis. De fato, as dimensões D1 e D2 foram automatizadas com a ajuda do detector *AZEA* (mais detalhes no Capítulo 6).

5.5 Considerações finais

Descrevemos sucintamente a evolução de sistemas de avaliação automática da qualidade de escrita. Citamos os problemas enfrentados que levaram pesquisadores a buscar alternativas automáticas para a avaliação automática. Descrevemos brevemente algumas das técnicas utilizadas nos sistemas de avaliação automática. Algumas técnicas mais complexas tiveram apenas seu funcionamento exemplificado. Focamos também em técnicas e sistemas que

avaliam estilo, uma vez que a análise automática do conteúdo não é aplicável em textos científicos, embora traços deste tipo de sistemas possam ser utilizados para personalizar um sistema de avaliação automática da qualidade de escrita específico para uma área, considerando os termos usuais mais adotados. Apresentamos também os resultados de avaliação desses sistemas por seus respectivos autores, sendo que uma avaliação comparativa é difícil por falta de um *cópus* comum para o teste da tarefa e eleição de métricas a serem empregadas na avaliação.

Por fim, descrevemos a rubrica desenvolvida para resumos científicos em inglês. Esta rubrica faz largo uso da estrutura esquemática do resumo. Exploramos o que cada uma de suas sete dimensões avalia, e indicamos quais necessitam da informação da estrutura esquemática do resumo a ser avaliado. Essa rubrica foi aplicada por diferentes avaliadores humanos, para medir a concordância e estimar a reprodutibilidade das dimensões propostas. Duas dimensões, D5 e D6, tiveram valores de *Kappa* baixos, o que indica baixa concordância entre os anotadores. A dimensão D5 que apresentamos já é a versão reformulada.

6 *Detecção automática da estrutura esquemática*

Neste capítulo, descrevemos vários experimentos para a tarefa de detecção automática da estrutura esquemática (EE) para resumos científicos em inglês. Vários detectores foram criados, seja pela variação de algoritmos de indução, do conjunto de atributos ou de abordagem de resolução. O melhor detector foi denominado **AZEA** (*Argumentative Zoning for English Abstracts*). Baseamos nosso trabalho na asserção de autores como Swales (1990) e Weissberg & Buker (1990) de que um texto científico obedece implicitamente uma estrutura de organização e exposição, que chamamos de estrutura esquemática. Estes detectores são baseados em aprendizado de córpus e na abordagem de segmentação por meio de “zonas argumentativas” proposta por (TEUFEL; MOENS, 2002), que foi descrita em detalhes no Capítulo 4. Muitas das decisões tomadas no desenvolvimento dos detectores foram amparadas pelas escolhas feitas na construção do *AZPort* (FELTRIM, 2004), descrito no Capítulo 4. Basicamente, os detectores têm em seu núcleo uma reimplementação do AZ (TEUFEL, 1999; TEUFEL; MOENS, 2002), com melhorias sugeridas na literatura e aprendidas através da experimentação.

Todo o processo de detecção da estrutura esquemática, mostrado na Figura 6, foi explorado. Entretanto, focamos na análise, construção e ajuste do conjunto de atributos utilizados para a representação de cada sentença e na aplicação e avaliação de diferentes algoritmos de aprendizado de máquina para a indução de classificadores capazes de usufruírem da representação que propomos. Foram testadas até mesmo diferentes ferramentas de PLN utilizadas no pré-processamento. O último experimento, entretanto, alterou a abordagem de resolução mostrada na Figura 6 e ganhou um nome especial: *Purpose Splitter*, pois a detecção é realizada em dois passos, começando da classificação do propósito de um resumo.

O objetivo do processo descrito na Figura 6 é o de identificar os componentes da estrutura esquemática de um resumo qualquer em língua inglesa. Pragmaticamente,

Figura 27: Estrutura do processo de detecção automática da estrutura esquemática

desenvolvemos um processo automático para rotular cada uma das sentenças de um resumo em língua inglesa com categorias descritas em esquema de anotação.

Este objetivo engloba a automatização de duas tarefas idênticas *a priori*, mas que se revelam distintas na prática. A primeira é a de identificar a estrutura esquemática de resumos supostamente bem escritos, como, por exemplo, resumos publicados em revistas científicas. A outra tarefa é a de acoplar a detecção automática da estrutura esquemática em uma ferramenta de escrita, e detectar a EE de um resumo que pode conter diversos erros não esperados em resumos publicados, tais como erros de ortografia ou mau uso de estruturas da língua, principalmente quando esses resumos são escritos por uma pessoa que não tenha o inglês como língua materna.

Este capítulo está dividido em seis seções. Na Seção 6.1 são descritas as etapas de pré-processamento, sendo que o esquema de anotação para resumos está descrito na Seção 6.2 e a definição e extração dos atributos utilizados são destacados na Seção 6.3. Na Seção 6.4, é tratada com mais profundidade a detecção dos componentes da estrutura esquemática de resumos. Apresentamos na Seção 6.5 uma proposta de extensão da metodologia desenvolvida para resumos para as outras seções de artigos científicos. Por fim, discutimos de maneira geral os experimentos realizados na Seção 6.6, indicando limitações e sugestões futuras de complementação dos experimentos realizados.

6.1 Pré-processamento do Cópus

O pré-processamento é formado por duas etapas. A primeira consiste em incorporar informações ou traços lingüísticos aos textos, basicamente separadas em três fases: determinação dos limites sentenciais, determinação das palavras (*tokens*) e incorporação da

sua classe/função morfofossintáticas a cada *token*. Esta primeira etapa é feita com o auxílio de ferramentas de PLN, descritas no Capítulo 3. O fluxo do pré-processamento, que chamamos de primeira etapa, é descrito a seguir:

1. **Sentenciador:** Divide o resumo em sentenças. O sentenciador que incorporamos ao AZEA foi o *OpenNLP*. O sentenciador não foi aplicado em nossos experimentos, visto que todos os resumos já estavam separados em sentenças.
2. **Tokenisador:** Divide as sentenças em *tokens*. O tokenisador utilizado foi o *script sed* disponibilizado no site do projeto *Penn Treebank*.
3. **Etiquetador Morfofossintático:** Foi utilizado o MXPOST para a tarefa de etiquetagem morfofossintática. O MXPOST requer que o texto de entrada esteja no formato sejam sentenças tokenizadas, o que é garantido pelo fluxo do pré-processamento.

No treinamento, é importante armazenar a informação das categorias dadas manualmente a cada sentença. São gerados dois arquivos, ambos com o mesmo número de linhas, que corresponde ao número total de sentenças. O primeiro contém as categorias de cada uma das sentenças do resumo, na ordem do fluxo do texto. O segundo, também na ordem do fluxo do texto, contém cada uma das sentenças, tokenizadas e com a etiqueta morfofossintática anexada ao final de cada *token*. Esse segundo arquivo é usado na extração dos valores dos atributos.

Na segunda etapa é construída uma tabela atributo-valor para cada texto. A descrição e como é feita a extração de cada atributo que compõe essa tabela atributo-valor é descrita na Seção 6.3. Na próxima seção descrevemos o esquema de anotação que usamos para classificar os resumos.

6.2 Esquema de anotação para resumos

As componentes esquemáticas, ou que chamaremos também de categorias, são os possíveis rótulos que serão dados a cada sentença. Utilizamos o esquema do SciPo-Farmácia, que é mostrado na Tabela 11.

O esquema que adotamos foca na detecção das seis componentes esquemáticas usadas no SciPo-Farmácia. Este esquema é muito similar ao esquema utilizado no *AZPort*

Tabela 11: Categorias usadas no *AZEA*

Contexto (Ct)
Lacuna (L)
Propósito (P)
Metodologia (M)
Resultado (R)
Conclusão (Co)

(FELTRIM, 2004). O *AZPort* contém todas as nossas componentes esquemáticas e difere apenas por incorporar uma componente denominada “Estrutura”, que, entretanto, é pouco representativa no *córpus* utilizado em seu treinamento do (2% das sentenças). Acreditamos que o valor de concordância mensurado pela estatística *Kappa* no esquema utilizado pelo *AZPort* seja muito similar, pois ambos tratam resumos científicos, embora difiram na língua em que esses estão escritos.

Não temos estudos sobre a reprodutibilidade do esquema exato que usamos. Usamos os estudos do *AZPort* como base. O valor do *Kappa* relatado no *AZPort* (FELTRIM, 2004) é de 0,69 (3 anotadores, 320 sentenças), para sete categorias. O valor de 0,69 indica que o sistema é reproduzível, segundo recomendações da literatura e a subjetividade da tarefa (FELTRIM, 2004). Além disso, o *Kappa* reportado é próximo do valor obtido em um experimento similar por Teufel et al. (1999), cujo valor de *Kappa* foi de 0,71 (4261 sentenças, 3 anotadores).

6.3 Atributos

A extração dos atributos é o último passo antes da classificação das sentenças. O *AZEA* baseia-se em algoritmos de aprendizado de máquina, e não trabalha com as sentenças puras, e sim com um vetor atributos.

Basicamente, os atributos do *AZEA* são uma reimplementação de alguns atributos do AZ (TEUFEL; MOENS, 2002; TEUFEL, 1999). Não foram implementados todos os atributos descritos Teufel & Moens (2002), e os atributos escolhidos são muito similares às escolhas feitas para o *AZPort* (FELTRIM, 2004). No total, o *AZEA* extrai 22 atributos de cada sentença, que são mostrados na Tabela 12.

Cada resumo gera uma tabela atributo-valor, que são vetores de atributos, sendo que cada sentença do resumo é representada por um vetor.

Tabela 12: Conjunto de atributos usados no *AZEA*

Atributo	Tipo	Descrição
Tamanho		Comprimento da sentença
Localização	Contextual	Posição da sentença no texto
Histórico		Categoria da sentença anterior
Tempo	Sintaxe	Tempo verbal do primeiro verbo finito da sentença
Modal	Verbal	Existência de auxiliar modal
Voz		Voz verbal
Expressões	Padrões Textuais	Primeira expressão-padrão contida na sentença
Formulaicas		
Agente		Primeira expressão-padrão, procurada no agente da sentença
<i>Abstract</i>		Primeira expressão-padrão de resumos contida na sentença
<i>Formulaic</i>		
Padrões		13 atributos, cada atributo mapeia um grupo de expressões indicativas
Booleanos		

6.3.1 Contextuais

São os únicos atributos que exploram o contexto externo aos limites sentenciais. Os atributos contidos nesse grupo extraem informações relativas à localização da sentença dentro do resumo, ou seja, buscam informações sobre o contexto no qual a sentença está inserida. Como já explicado neste trabalho, nos apoiamos nos autores que defendem que textos científicos compartilham uma estrutura comum: a estrutura esquemática. As componentes dessa estrutura esquemática, as quais propomos identificar automaticamente, não são arrançadas caoticamente, e sim seguem uma ordem convencional. Sendo assim, os atributos contextuais exploram essa estrutura implícita dos textos científicos, aproveitando-se de organizações comuns e restrições de associação de componentes. Dois atributos representam este grupo:

1. **Localização:** Marca a posição relativa da sentença em relação ao texto. Esse atributo também é explorado em outras tarefas e outros gênero. Por exemplo, na sumarização de textos jornalísticos, as primeiras sentenças costumam ser as mais informativas. No caso dos *abstracts*, as primeiras sentenças costumam ser ou da categoria *Contexto* ou da categoria *Propósito*¹. Já na última sentença, não são esperadas sentenças das categorias *Contexto* ou *Propósito*².

Valores (5): *primeira, segunda, mediana, penúltima, última.*

¹ *Abstracts* que começam diretamente com *Propósito* tendem a ser curtos (comum em *abstracts* resumidos, que trazem somente os componentes essenciais).

² Em nossos corpus anotados manualmente, não existe um caso sequer.

Extração: Os valores são intuitivos, mas há uma pequena confusão em texto pequenos, com menos de 5 sentenças. Para evitar situações ambíguas, o algoritmo usa a seguinte ordem de prioridades:

- (a) marca a primeira sentença do resumo com *primeira*;
- (b) se a última sentença do resumo não está marcada, marca-a com *última*;
- (c) se a segunda sentença do resumo não está marcada, marca-a com *segunda*;
- (d) se a penúltima sentença do resumo não está marcada, marca-a com *penúltima*;
- (e) marca todas as sentenças ainda não marcadas com *mediana*.

2. **Histórico:** Recorda a categoria da sentença anterior. Explora o fluxo das componentes no decorrer do texto. Os autores citados no Capítulo 2 defendem que existe um arranjo implícito das componentes, algo análogo a uma “coesão da estrutura esquemática”. Por exemplo, se uma sentença da categoria *Lacuna* é introduzida, espera-se que a seguir seja anunciado o *Propósito* do trabalho. Esse atributo é de grande importância para todos os trabalhos correlatos descritos no Capítulo 4.

Valores (7): *contexto, lacuna, propósito, metodologia, resultados, conclusão* ou *nenhum*.

Extração: Se a sentença for a primeira do texto, recebe o valor *nenhum*. Caso contrário, recebe a categoria dada à sentença anterior à sentença em atual classificação. Na fase de teste é utilizada a categoria provável da sentença anterior. Quando um texto é submetido para o AZEA, este classifica-o da primeira sentença à última, uma de cada vez. A categoria dada pelo AZEA à sentença é então utilizada como o valor do atributo *Histórico* da próxima sentença.

Como exemplo, temos na Tabela 13 um resumo com 7 sentenças. Na primeira coluna, temos a categoria de cada uma dessas sentenças. Na segunda e terceira coluna são mostrados os valores dos atributos *Localização* e *Histórico*.

Tabela 13: Exemplo de extração de atributos contextuais *Localização* e *Histórico*

Categoria	<i>Localização</i>	<i>Histórico</i>
<i>Contexto</i>	<i>primeira</i>	<i>nenhum</i>
<i>Lacuna</i>	<i>segunda</i>	<i>contexto</i>
<i>Propósito</i>	<i>mediana</i>	<i>lacuna</i>
<i>Metodologia</i>	<i>mediana</i>	<i>propósito</i>
<i>Resultado</i>	<i>mediana</i>	<i>metodologia</i>
<i>Conclusão</i>	<i>penúltima</i>	<i>resultado</i>
<i>Conclusão</i>	<i>última</i>	<i>conclusão</i>

6.3.2 Sintaxe Verbal

Atributos sintáticos para a categorização de textos de diferentes gêneros têm sido amplamente estudados. Esses atributos também têm sido muito explorados no estudo de textos científicos. Diferentes autores utilizam-se de atributos sintáticos para descrever peculiaridades de gêneros científicos. Graetz (apud SWALES, 1990) inicia sua descrição da linguagem utilizada em *abstracts* com: “*The abstract is characterized by the use of past tense, third person, passive, and the non-use of negatives*”.

Estes atributos não só estão ligados ao gênero, mas também a seções dentro de um gênero. Swales (1990) faz distinções do uso de tempos verbais e do emprego da voz ativa em diferentes seções de um artigo científico, além de citar diversos outros estudos sobre gêneros nos quais estes atributos são utilizados.

Os atributos sintáticos são observados por meio de três atributos no AZEA: *Tempo*, *Voz* e *Modal*. Para a extração desses atributos, fixamos como ponto inicial de análise o *primeiro verbo finito*³ encontrado na sentença. O contexto da extração desses atributos concentra-se nos *tokens* ao redor do verbo finito, sempre contendo-se dentro dos limites sentenciais. A definição dos valores dos atributos é feita pela análise das etiquetas morfossintáticas atribuídas a cada um dos *tokens* e da aplicação de regras gramaticais. Os três atributos são descritos abaixo:

1. **Tempo:** Determina o tempo verbal do primeiro verbo finito da sentença.

Valores (17): *BaseForm*, *Fut*, *FutCont*, *FutPerf*, *FutPerfCont*, *Gerund*, *Past*, *PastCont*, *PastPart*, *PastPerf*, *PastPerfCont*, *Pres*, *PresCont*, *PresPerf*, *PresPerfCont*, *Would* e *noverb*.

Extração: Achado o verbo finito da sentença, olha o contexto do verbo em busca de verbos auxiliares, ou seja, modificadores. Marcações temporais morfológicas⁴ são dadas pelas etiquetas morfossintáticas.

2. **Modal:** Verifica se o verbo está sendo modificado por um auxiliar modal. Exemplos de auxiliares modais são *could*, *should*, *will*, entre outros.

Valores (3): *modal*, *nomodal* e *noverb*.

³Entende-se por *verbo finito* como sendo um verbo que não esteja no infinitivo

⁴Marcações de gerúndio, passado simples e particípio são observadas por meio de seus traços morfológicos

Extração: Dado o verbo finito, verifica se existe um *token* cuja a etiqueta seja um auxiliar modal⁵. A busca é feita até três *tokens* à esquerda do verbo finito⁶. Se a busca parar sem encontrar um auxiliar modal, o valor do atributo será *nomodal*.

3. **Voz:** Determina se uma sentença está na voz passiva ou ativa. Esse atributo também é utilizado na extração do atributo *Agente*, descrito ulteriormente na Seção 2.

Valores (3): *ativa*, *passiva* ou *noverb*.

Extração: Examina se o primeiro verbo finito está na forma passiva, da forma *verbo To Be + Participípio*. Se for dessa forma, recebe *passivo*, ou *ativo* caso contrário.

Se nenhum verbo finito é encontrado, cada um dos três atributos sintáticos descritos recebem o valor *noverb*.

Os valores desses atributos não foram analisadas manualmente. Portanto, não sabemos com qual precisão cada um dos processos é executado. A extração dos atributos sintáticos é crítica, e depende totalmente da precisão da etapa de pré-processamento: um erro na etiquetagem morfosintática ou na definição do verbo finito compromete a extração de todos os atributos sintáticos.

6.3.3 Padrões Textuais

Grande parte dos atributos utilizados pelo AZEA procuram por expressões-padrão no texto. A anotação humana de uma componente esquemática é alcançada por meio da identificação de marcadores ou expressões comumente utilizadas para tal fim. Expressões tais como “*the aim of this research is*” ou “*in this paper*” usualmente indicam que as sentenças que as possuem representam a categoria *Propósito*. Os sistemas de apoio à escrita científica descritos no Capítulo 2, (Seções 2.2 e 2.3) utilizam estas expressões como estruturas reutilizáveis, para que o usuário reaproveite texto e apenas altere as informações factuais, relativas ao trabalho em questão. Os atributos são descritos abaixo:

1. **Formulaic Patterns:** Busca expressões-padrão na sentença. As expressões-padrão estão agrupadas, sendo que cada grupo define um tipo. Os tipos e as expressões-padrão utilizadas estão descritos no Apêndice D.1 em (TEUFEL, 1999).

⁵Em nosso esquema de anotação, a etiqueta que descreve auxiliares modais é MD

⁶Se algum token na busca não estiver marcado com as etiquetas PRP, RB ou EX, pára a busca

Valores (20): *textstructure, similarity, contrast, comparison, bad, them, continue, future, good, general, detail, affect, no_textstructure, method, us_previous, in_order_to, here, gap, tradition, none.*

Extração: Percorre a sentença do primeiro ao último *token*, tentando reconhecer alguma expressão-padrão. A primeira expressão encontrada determina o valor do atributo e encerra a busca. Se nenhuma expressão-padrão é encontrada, recebe *none*.

2. **Agente:** Busca por expressões-padrão no agente das orações. Considera o sujeito da oração se esta está na voz ativa, ou o sujeito da passiva se a oração está na voz passiva. As expressões-padrão usadas estão descritas no Apêndice D.2 em (TEUFEL, 1999).

Valores (14): *ref, solution, us, textstructure, us_previous_selfcite, gap, problem, them_pronoun, general, aim_ref, ref_us, them_cite, our_aim, none.*

Extração: Para cada verbo finito da sentença, verifica se este está na voz passiva ou ativa (usando o processo descrito na Seção 6.3.2). Se a “oração” estiver na voz passiva, busca por expressões-padrão de *Agente* à direita do verbo; busca à esquerda caso estiver na voz ativa. A busca por expressões-padrão tenta não ultrapassar limites oracionais, e pára quando encontra um outro verbo finito ou uma pontuação. O tipo da primeira expressão-padrão encontrada é dado como valor do atributo.

Esses atributos, *Formulaic patterns* e *Agentes*, são retirados no trabalho de Teufel (1999). Alguns padrões acabaram não sendo utilizados em nossos atributos por dois motivos: valiam-se de informações que não utilizamos (como é o caso de padrões que envolvem o reconhecimento de trechos de citações) ou pela falta de especificação de partes do padrão (basicamente referência a grupos de palavras não detalhados em (TEUFEL, 1999)).

6.3.3.1 Expressões-padrão de *abstracts*

No *AZPort*, as expressões-padrão são condensadas em um único atributo. As categorias, que são os valores desse atributo, são representadas por um conjunto de expressões-padrão. Esse atributo obteve a maior contribuição individual nesse classificador (FELTRIM, 2004).

Como parte de um estudo lingüístico, as sentenças dos 43 *abstracts* do SciPo-Farmácia foram analisadas em busca de expressões-padrão. Desse estudo, feito por uma lingüista⁷,

⁷Carmen Dayrell, bolsista de pós-doutorado no NILC.

foram elaborados 14 atributos. O atributo *Abstract Formulaic* assemelha-se ao atributo *Formulaic expressions*. Os outros 13 atributos mapeiam grupos de expressões indicativas. São denominados booleanos, pois detectam se pelo menos uma das expressões-padrão do atributo é encontrada na sentença, e não contabilizam quantas expressões são encontradas.

1. ***Abstract Formulaics***: Busca por expressões-padrão específicas para *abstracts*.

Valores (5): *comparison, negative_pass, expressions, active_voice* ou *none*.

Extração: Idêntica a extração do atributo *Formulaic Patterns*.

2. **Padrões booleanos**: Diferentemente dos outros atributos de busca por expressões-padrão, os padrões booleanos não tem um tipo. Eles são desmembrados em 13 atributos, sendo que cada um deles relata a ocorrência de pelo menos uma expressão-padrão. Esses atributos são chamados dessa maneira pois não diferenciam qual tipo de padrão foi encontrado ou quantas vezes padrões da mesmo atributo foram encontrados.

Atributos (13): *analyse, carried_out, conclude, confirm, contribute, examine, find, here, investigate, obs, verbs1, verbs2, verbs_variation*.

Valores (2): “*nome do próprio atributo*” ou *N*.

Extração: Para cada atributo, busca as expressões-padrão por toda a sentença. Se alguma é encontrada, pára e retorna o nome do atributo; se nada é encontrado retorna *N*.

6.3.3.2 Formato das expressões

Aqui descrevemos como podem ser formuladas as expressões-padrão. Uma expressão-padrão é uma seqüência de *tokens*. Os *tokens* de uma expressão-padrão podem ser de quatro tipos:

1. **palavra_etiqueta**: obriga que o par *token_etiqueta* sendo comparado seja idêntico ao da expressão.
2. **palavra**: apenas o *token* sendo comparado deve ser idêntico ao da expressão.
3. **etiqueta**: apenas a *etiqueta* sendo comparada deve ser idêntica à da expressão. A etiqueta deve estar em letras maiúsculas, para diferenciar-se do *token*.

4. **@Conceitos:** é um conjunto expressões, que podem ser compostas pelos *tokens* descritos acima. Não agrega nenhum tipo diferente de comparação, é apenas utilizado para agrupar conceitos e facilitar na descrição das expressões.

6.3.4 Comprimento da sentença

O comprimento de sentença é uma medida muito simples, mas que em alguns casos pode ser crucial ajuda. Sentenças mais longas costumam ser mais complexas. Este atributo conta os *tokens* da sentença, descartando os *tokens* de pontuação.

Valores(3): *small, med, big*.

Extração: Conta o número de *tokens* da sentenças, desprezando os *tokens* de pontuação.

Neste trabalho, as sentenças com até 20 *tokens* recebem *small*, entre 21 e 40 *tokens* recebem *med* e acima de 41 recebem *big*.

6.4 Treinamento e avaliação do *AZEA*

Nesta seção, apresentamos a avaliação da detecção automática dos componentes da estrutura esquemática de resumos científicos. Nas seções anteriores deste capítulo, descrevemos como mapear cada sentença de um resumo em um conjunto de atributos, descrito na Seção 6.3. A metodologia deste trabalho é baseado em aprendizado de máquina supervisionado, auxiliado por *corpus*. Três *corpus* são utilizados nesta seção, sendo um para o treinamento e dois para teste. A indução de classificadores foi feita por meio das implementações de algoritmos de aprendizado supervisionado do WEKA. Os algoritmos escolhidos para indução dos classificadores foram: SMO, *Naive Bayes*, J48 e JRip. Esses quatro algoritmos, descritos com mais detalhes no Capítulo 3, são algoritmos amplamente utilizados em tarefas de *Text Mining* e de Categorização Textual (SEBASTIANI, 2002; GONÇALVES, 2002; JOACHIMS, 1998). Quando mencionada nesta seção, a *baseline* refere-se a um classificador cuja a estratégia é rotular toda instância com a classe mais freqüente do *corpus* utilizado no treinamento. O método usado na estimativa do erro real utilizado nos treinamentos foi o *10-fold stratified cross-validation*.

6.4.1 Córpus

6.4.1.1 Córpus de treinamento

Para o treinamento e os testes do detector automático da EE, vários córpus foram utilizados. No treinamento, inicialmente foi utilizado um córpus com 43 resumos científicos em inglês da área de Ciências Farmacêuticas, presente na ferramenta SciPo-Farmácia, descrito na Seção 3.2 do Capítulo 3. Este córpus foi anotado por um lingüista e revisado por 2 especialistas em escrita científica. Foram acrescentados 31 resumos publicados ao córpus, obtidos na Internet, cuja estrutura esquemática foi anotada por um especialista em escrita científica, totalizando 74 resumos no córpus. Detalhes sobre o tamanho do córpus de treino podem ser analisados em mais detalhes na Tabela 14.

Tabela 14: Estatísticas lexicais do córpus de treinamento, entre parênteses, temos o desvio padrão

	Tamanho do córpus	Média	Maior Resumo	Menor Resumo
Em palavras	15207	205 (47)	466	107
Em sentenças	622	8,4 (2,2)	18	4

A categoria *Resultado* é a mais freqüente (44,8%), enquanto que a categoria *Lacuna* é a mais escassa (2,1%). Na Tabela 15 são mostradas algumas estatísticas em relação à distribuição das sentenças e resumos com relação às componentes esquemáticas. Na primeira coluna é mostrada a porcentagem de resumos cuja a categoria que encabeça a linha aparece pelo menos uma vez (o número de textos nos quais esta condição é satisfeita está entre parênteses), e na segunda coluna está a porcentagem da categoria no córpus, em relação ao número total de sentenças (o número de sentenças de cada categoria está entre parênteses).

Tabela 15: Distribuição das categorias no córpus de treinamento

Categoria	Presença da categoria	Frequência no córpus
<i>Contexto</i>	56,7% (42)	11,6% (72)
<i>Lacuna</i>	14,8% (11)	2,1% (13)
<i>Propósito</i>	98,6% (73)	12,5% (78)
<i>Metodologia</i>	70,3% (52)	18% (112)
<i>Resultado</i>	100% (74)	44,8% (279)
<i>Conclusão</i>	78,4% (58)	11% (68)

Todos os resumos foram anotados manualmente, sendo que cada sentença está rotulada com uma das categorias esquemáticas da Tabela 11. Os textos estão codificados com etiquetas XML, como exemplificado na Figura 28.

```

- <Abstract id="ab_01">
- <Referencia>
  http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=14990354&itool=iconabstr
  Inhibition of copper-catalyzed cysteine oxidation by nanomolar concentrations of iron salts Munday R, Munday CM, Winterbourn
  CC.
</Referencia>
+ <Subcomponente></Subcomponente>
+ <Subcomponente></Subcomponente>
- <Subcomponente>
  <Nome>Propósito</Nome>
  - <Estratégia>
    <Nome>Apresentar o propósito com os resultados</Nome>
    In the present study,
    <Marcador> however </Marcador>
    , the rate of copper-catalyzed oxidation of cysteine was found to be increased by buffer purification with Chelex resin or by
    addition of micromolar concentrations of the specific iron chelator desferrioxamine (DFO).
  </Estratégia>
</Subcomponente>

```

Figura 28: Parte da estrutura XML do resumo “01” do corpus do SciPo-Farmácia

Nesta figura são mostradas as informações atachadas a cada uma das sentenças do resumo. Cada etiqueta *Subcomponente* é composta pelas etiquetas *Nome* (componente esquemática) e *Estratégia*, sendo que a sentença está contida nesta última. Palavras da sentença podem ser anotados, como é o caso da palavra *however*, anotada com a etiqueta *Marcador*.

6.4.1.2 Corpus de teste

Além do corpus usado no treinamento, foram utilizados dois outros corpus para testes, ambos com a estrutura esquemática anotada manualmente: **Atípicos** e **Alunos**. Como já mencionado, o corpus de treinamento contém apenas resumos cuja a estrutura esquemática se enquadra nas suposições descritas no Capítulo 2. Esses corpus são de grande importância para a cobertura dos testes, visto que englobam os dois alvos do módulo de detecção automática da estrutura esquemática: reconhecimento da estrutura em textos publicados (corpus de treinamento e corpus *Atípicos*) e ajuda na escrita de um resumo científico (corpus *Alunos*).

O corpus *Atípicos* é formado por 18 resumos (121 sentenças, 2880 palavras) publicados da área de Ciências Farmacêuticas, cuja a estrutura esquemática foi considerada *atípica*, segundo os critérios descritos em (SWALES, 1990; WEISSBERG; BUKER, 1990). Usamos este corpus pois são resumos publicados, e pretendemos avaliar nosso modelo sobre resumos sem a estrutura esquemática convencional.

Tabela 16: Distribuição das categorias no corpus *Atípicos*

Categoria	Presença da categoria	Frequência no corpus
<i>Contexto</i>	33,3% (6)	9,1% (11)
<i>Lacuna</i>	0% (0)	0% (0)
<i>Propósito</i>	100,0% (18)	17,4% (21)
<i>Metodologia</i>	72,2% (13)	19,8% (24)
<i>Resultado</i>	77,8% (14)	41,3% (50)
<i>Conclusão</i>	55,6% (10)	12,4% (15)

O corpus *Alunos*, também com 18 resumos (143 sentenças, 3051 palavras) é formado por textos de alunos de um curso de escrita científica em inglês. Na Tabela 17 são mostradas as estatísticas desse corpus. A distribuição das sentenças por categoria desse corpus difere do corpus de treinamento, sendo a categoria *Contexto* a mais frequente (36,4%).

Tabela 17: Distribuição das categorias no corpus *Alunos*

Categoria	Presença da categoria	Frequência no corpus
<i>Contexto</i>	100,0% (18)	36,4% (52)
<i>Lacuna</i>	22,2% (4)	4,2% (6)
<i>Propósito</i>	83,3% (15)	12,6% (18)
<i>Metodologia</i>	61,1% (11)	16,1% (23)
<i>Resultado</i>	77,8% (14)	23,1% (33)
<i>Conclusão</i>	55,6% (10)	7,7% (11)

6.4.2 Avaliação

Primeiramente, foram testados diferentes tipos de algoritmos de AM supervisionado, encontrados no WEKA. Quatro algoritmos, descritos no Capítulo 3, foram escolhidos: *Naive Bayes* (estatístico, bayesiano), J48 (árvore de decisão), JRip (indutor de regras) e o SMO (estatístico, baseado em SVM). A não ser que seja especificado de maneira diferente, assume-se que os dados de avaliações reportados sobre o corpus de treinamento utilizam a estima de erro real *10-fold stratified cross-validation*. Sempre que citada, a *baseline* refere-se a um classificador que rotula qualquer novo caso com a classe majoritária do corpus de treinamento. Em nosso corpus de treinamento, a categoria majoritária é *Resultado*.

O primeiro experimento avaliou o impacto na taxa de acerto e na estatística *Kappa* para diferentes algoritmos de indução de classificadores. Na Tabela 18 é mostrada a comparação entre os diferentes tipos de algoritmos utilizados, juntamente com a *baseline*.

Analisando o valor de *Kappa* dos algoritmos, percebemos que todos obtiveram ótimos valores, considerando a subjetividade da tarefa. Em um experimento sobre a

Tabela 18: Variando os algoritmos de AM supervisionado

	SMO	Naive Bayes	J48	JRip	Baseline
Acerto	80,39%	77,97%	77,17%	77,81%	44,86%
Kappa	0,73	0,69	0,69	0,69	0
<i>Macro-F</i>	0,78	0,70	0,74	0,74	0,10

reprodutibilidade do esquema de anotação do *AZPort* (FELTRIM, 2004), o valor de *Kappa* entre os três anotadores humanos é de 0,71, e considerando a similaridade entre o esquema de anotação do *AZPort* e o nosso, os valores obtidos e os relatados são muito próximos.

Entretanto, os modelos induzidos sobre o corpus de treinamento não obtiveram o mesmo êxito na classificação dos resumos dos corpus de teste. Na Tabela 19, temos os resultados da aplicação dos modelos induzidos sobre os corpus de testes.

Tabela 19: Aplicando o *AZEA* sobre os corpus de teste

Corpus	Medida	SMO	Naive	J48	JRip
Atípicos	Acerto	61.98%	67.77%	62.81%	64.46%
	Kappa	0.48	0.55	0.50	0.52
	<i>Macro-F</i>	0.49	0.54	0.50	0.51
Alunos	Acerto	74.83%	73.43%	79.02%	60.14%
	Kappa	0.67	0.64	0.73	0.51
	<i>Macro-F</i>	0.70	0.64	0.76	0.56
Atípicos + Alunos	Acerto	68.94%	70.83%	71.59%	62.12%
	Kappa	0.60	0.62	0.64	0.52
	<i>Macro-F</i>	0.64	0.63	0.68	0.55

O ranqueamento dos classificadores induzidos por diferentes algoritmos não é estável, ou seja, só da análise das métricas não é possível eleger um indutor cuja a aplicação seja ótima em todos os momentos. O classificador induzido pelo SMO, que tivera o melhor desempenho na fase de treinamento, não manteve a primeira posição nos testes. No corpus *Atípicos*, o melhor desempenho foi alcançado pelo classificador induzido pelo *Naive Bayes*. Entretanto, o classificador induzido pelo J48 obteve o melhor desempenho nos testes com o corpus *Alunos*, e, na média, saiu-se melhor nos testes sobre os dois corpus. Outro ponto importante a ser ressaltado são os valores de *Kappa* nos testes. No corpus *Alunos* esses valores foram bons (entre 0,67 e 0,73), com exceção do JRip (0,51). Contudo, no corpus *Atípicos* esses valores variam entre 0,48 e 0,55, muito abaixo dos valores obtidos no treinamento (entre 0,69 e 0,73).

Uma justificativa para essa queda no acerto deve-se à importância dos atributos contextuais na classificação. Na Tabela 20 são mostrados os valores de *Kappa* como uma estimativa da contribuição individual dos atributos, para os quatro classificadores induzidos por diferentes algoritmos. Nas últimas quatro linhas dessa tabela, são mostradas

as contribuições de alguns grupos de atributos. Os resultados foram obtidos no cópup de treinamento.

Tabela 20: Contribuição dos atributos. Valores do Kappa

Atributo	Atributo(s) Excluído				Atributo(s) Sozinho			
	SMO	Naive	J48	JRip	SMO	Naive	J48	JRip
<i>Posição</i>	0,66	0,63	0,65	0,60	0,39	0,39	0,41	0,32
<i>Histórico</i>	0,64	0,62	0,64	0,60	0,42	0,42	0,41	0,23
<i>Tamanho</i>	0,72	0,70	0,69	0,68	0,00	0,00	0,00	0,00
<i>Formulaic Patterns</i>	0,71	0,70	0,69	0,66	0,07	0,08	0,07	0,07
<i>Agente</i>	0,74	0,70	0,68	0,65	0,15	0,15	0,15	0,15
<i>Modal</i>	0,72	0,70	0,69	0,66	0,05	0,05	0,05	0,01
<i>Tempo</i>	0,71	0,69	0,66	0,67	0,11	0,11	0,11	0,05
<i>Voz</i>	0,73	0,70	0,71	0,70	0,12	0,12	0,12	0,00
<i>Abstract Formulaic</i>	0,71	0,70	0,65	0,66	0,14	0,14	0,14	0,14
<i>Bool Patterns</i>	0,68	0,67	0,70	0,67	0,15	0,16	0,15	0,11
<i>Contextuais</i>	0,44	0,40	0,37	0,26	0,57	0,55	0,57	0,50
<i>Sintaxe Verbal</i>	0,71	0,70	0,68	0,66	0,19	0,20	0,15	0,05
<i>Expressões-Padrão</i>	0,66	0,65	0,64	0,62	0,22	0,22	0,24	0,22

Os atributos que mais ajudam para a classificação são os atributos contextuais, sendo que o atributo *Histórico* é o atributo com maior capacidade de predição. Na segunda coluna temos o valor do *Kappa* quando o atributo é desconsiderado, e na terceira coluna o *Kappa* para o classificador somente com o atributo indicado na linha. As últimas quatro linhas dessa tabela mensuram a contribuição de grupos de atributos, tais como foram divididos na Seção 6.3.

Por fim, mostramos nas Tabelas 21 e 22 as métricas de avaliação de classe e a matriz de confusão para o classificador induzido com o algoritmo SMO sobre o corpus de treino.

Tabela 21: Métricas estatísticas de avaliação de cada classe do *AZEA* usando SMO sobre cópup de treino

Categoria	Precisão	Cobertura	Medida-F
Ct	77,8%	77,8%	77,8%
L	80%	61,5%	69,6%
P	82,1%	82,1%	82,1%
M	69,4%	76,8%	72,9%
R	85,6%	83,2%	84,4%
Co	82,1%	80,9%	81,5%

A maioria das categorias mantém um valor de medida-F em torno da taxa de acerto do classificador (80,4%), variando entre 77,8% (*Contexto*) e 84,4% (*Resultado*). Duas categorias com pior desempenho, segundo a medida-F, são *Lacuna* e *Metodologia*. *Lacuna* obteve

o menor desempenho (69,6% de medida-F). Atribuímos este problema ao baixo número de sentenças dessa categoria, apenas 13 casos no corpus de treinamento. Já a categoria *Metodologia* remete à pior precisão por classe desse classificador, com 69,4%. Essa categoria é confundida com as sentenças da categoria *Resultado* pelo classificador, o que é revelado quando observamos a matriz de confusão na Tabela 22.

Tabela 22: Matriz de confusão para o AZEA

	Ct	L	P	M	R	Co
Ct	56	2	<u>11</u>	2	0	1
L	5	8	0	0	0	0
P	<u>10</u>	0	64	1	3	0
M	1	0	1	86	<u>24</u>	0
R	0	0	2	<u>34</u>	232	<u>11</u>
Co	0	0	0	1	<u>12</u>	55

De fato, ao analisarmos a matriz de confusão na Tabela 22, notamos que esse classificador confunde alguns pares de categorias. O par que gera maior confusão é composto pelas categorias *Metodologia X Resultado* (47% dos erros), seguido dos pares *Resultado X Conclusão* (19%) e *Contexto X Propósito* (17%). Se agruparmos as categorias *Metodologia* e *Resultado*, o erro do classificador diminui para 9,8% (Kappa 0,83, baseado no SMO). Discernir entre estas duas categorias foi considerado árduo em alguns momentos pelo classificador humano. Normalmente, ocorre que orações tanto de *Resultado* quanto de *Metodologia* estão arrançadas em uma mesma sentença.

Em geral, os mesmos pares de classes são confundidos quando os classificadores foram aplicados às sentenças dos corpus de testes. Na Tabela 23, temos as matrizes de confusão para os corpus (1) *Alunos* e (2) *Atípicos*, treinados com os algoritmos *Naive Bayes* e J48, respectivamente. Em (1), os erros concentram-se na categoria *Contexto* (33% dos erros), e o par *Contexto X Propósito* registra 9 dos 30 erros (30%) cometidos pelo classificador. Na matriz de (2), novamente o par *Metodologia X Resultado* mantém-se no topo, com 17 dos 39 erros (43,5% dos erros).

6.4.3 *Purpose Splitter*

Uma outra maneira que encontramos para aprimorar o classificador da EE de resumos foi utilizar uma abordagem de decomposição da tarefa em dois classificadores. Um deles é repomssável por anotar a sentença de propósito, pois esta sentença é considerada um divisor de categorias. As sentenças das categorias *Contexto*, *Lacuna* tendem a anteceder a primeira

Tabela 23: Matriz de confusão para a classificação dos corpúsculos de teste

	Ct	L	P	M	R	Co			Ct	L	P	M	R	Co
Ct	45	1	5	0	0	1		Ct	9	1	1	0	0	0
L	1	3	1	1	0	0		L	0	0	0	0	0	0
P	4	0	14	0	0	0		P	6	0	11	1	3	0
M	3	0	0	18	2	0		M	1	0	0	10	13	0
R	2	0	2	5	23	1		R	1	0	0	4	43	2
Co	0	0	0	0	1	10		Co	0	0	0	3	3	9
(1) <i>Alunos</i> X J48								(2) <i>Atípicos</i> X Naive Bayes						

sentença da categoria *Propósito*, enquanto que predominam as categorias *Metodologia*, *Resultado* e *Conclusão* ulteriormente. Essa organização das sentenças é constatada em nosso corpúsculo de treinamento, como podemos observar na Tabela 24.

Tabela 24: Ocorrência das categorias em relação à primeira sentença da categoria *Propósito*

Categoria	Antes do propósito	Depois do propósito
Contexto	69	3
Lacuna	13	–
Metodologia	3	109
Resultado	4	275
Conclusão	–	68

Para aproveitar esse fato, incluímos um atributo contextual, que nos fornece a posição da sentença em relação à primeira sentença de propósito do resumo, que descrevemos em seguida:

Purpose Position: Posição da sentença em relação à primeira sentença da componente *Propósito*. Usa um classificador auxiliar para extraí-la. Esse classificador auxiliar é binário, e determina se uma sentença deve ser rotulada como sendo da componente *propósito* ou de *outra componente*. Esse atributo pode assumir 3 valores:

- **purpose** para todas as sentenças marcadas como *propósito* por esse classificador auxiliar;
- **before_purpose** para todas as sentenças antes da primeira sentença rotulada *propósito* por esse classificador;
- **after_purpose** para todas as outras sentenças ainda não marcadas.

Os passos desta nova abordagem para a detecção da estrutura esquemática de um resumo utilizando esse atributo são:

1. Determinar as sentenças da categoria propósito do resumo

Se nenhuma sentença for rotulada com a categoria *Propósito*, desconsidera esse atributo.

2. Calcular o valor do atributo *PurposePosition* e agregá-lo ao conjunto de atributos

O melhor cenário para esta abordagem é quando a determinação do atributo *PurposePosition* é perfeita, o que implicaria termos um classificador que nunca erra ao classificar uma sentença de propósito. Neste caso, os classificadores atingiriam os valores descritos na Tabela 25. Entre parênteses, ao lado de cada estatística, estão as melhoras em relação ao *AZEA* sem esse atributo, relatado anteriormente na Tabela 18.

Tabela 25: Estatísticas dos classificadores supondo que a extração do atributo *PurposePosition* fosse ótima

Algoritmo	T. Acerto	Kappa	Macro-F
J48	82,32%(+5,31%)	0,76(+0,07)	80%(+7%)
JRip	82,15%(+4,34%)	0,76(+0,07)	81%(+7%)
Naive Bayes	83,28%(+5,31%)	0,77(+0,08)	76%(+6%)
SMO	84,73%(+4,18%)	0,79(+0,06)	84%(+6%)

Esse atributo ajuda a melhorar a tarefa se garantirmos que a precisão da escolha das sentenças da categoria *Propósito* seja próxima de 100%. Temos que anular os casos de “falsos positivos”, ou seja, que sentenças de outras categorias sejam classificadas como *Propósito*. Na Tabela 26 temos dados de classificadores que objetivam determinar se uma sentença é da categoria *Propósito* ou não. Nessa tabela podemos observar que a *baseline* é alta (87,5%), e a precisão da categoria *Propósito* não acompanha os valores da taxa de acerto do classificador.

Tabela 26: Classificadores binários de identificação de sentenças da categoria *Propósito*

	J48	JRip	Naive Bayes	SMO	Baseline
Acerto	91,5%	92,4%	93,1%	92,9%	87,5%
Kappa	0,58	0,66	0,66	0,66	0
Macro-F	0,79	0,83	0,83	0,83	0,47
Precisão(<i>Propósito</i>)	69,8%	69,1%	77,8%	74,3%	0%

A melhora quando se usa o atributo *PurposePosition*, na prática, é ínfima. Em nossos testes com os corpú de teste, não houve melhora. Para garantirmos que o atributo fosse determinado com precisão, treinamos um classificador bayesiano e apenas os exemplos rotulados com a classe *Propósito* com probabilidade superior a 0,95 foram considerados. O relaxamento dessa restrição acarreta aumento de erros, e o atributo passa a atrapalhar ao invés de ajudar. Nos testes com os corpú de resumos com estrutura atípica e de resumos de

alunos, apenas 4 dos 36 resumos tiveram suas sentenças da categoria *Propósito* identificadas, e o resultado da classificação foi o mesmo da classificação sem esse atributo.

Embora possa ser uma boa abordagem para a anotação humana, a separação entre as categorias que ocorrem antes do propósito e as que ocorrem depois dele não é o maior desafio da classificação automática. O classificador confunde-se nas categorias dentro de cada grupo. Se observarmos a matriz de confusão na Tabela 22, podemos notar que quase metade dos erros (58 de 121) são devidos à confusão entre as categorias *Metodologia* e *Resultado*. Os outros dois grupos de confusão, embora bem menores, são *Resultado* e *Conclusão* (23 erros) e *Contexto* e *Propósito* (21 erros). Ou seja, 84% dos erros cometidos pelo nosso classificador ocorre na confusão entre duas categorias, que são tipicamente encontradas ou ambas antes ou ambas depois da primeira sentença da categoria *Propósito*.

6.4.4 Comparação com outros sistemas

A comparação entre sistemas de detecção automática da estrutura esquemática não é uma tarefa fácil. Entre os problemas, ressaltamos os esquemas de anotação diferentes, indisponibilidade de uso do sistema para testes e dependência de língua. Entre os sistemas baseados em *Zonas Argumentativas*, o *AZ* não está disponível, e o *AZPort* é para o português. Para esses sistemas apenas temos os valores relatados por seus autores, que podem ser vistos na Tabela 27.

Tabela 27: Comparação dos sistemas baseados em *Zonas Argumentativas*. Valores relatados pelos autores

Sistema	Acerto	Kappa	Tarefa
<i>AZ</i> (TEUFEL; MOENS, 2002)	73%	0,45	Artigos científicos em inglês, 7 categorias
<i>AZPort</i> (FELTRIM, 2004)	72%	0,65	Resumos científicos em português, 7 categorias
<i>AZEA</i> (SMO)	80%	0,73	Resumos científicos em inglês, 6 categorias

6.4.4.1 Comparação com o mesmo corpus de treinamento

O único sistema com o qual o *AZEA* foi confrontado foi o *Mover*. Nessa comparação, foram utilizados apenas os 43 resumos contidos no SciPo-Farmácia. Nesta comparação, não foi usado o método de estimativa do erro real *k fold cross-validation*, pois o uso do *Mover* (tanto treinamento quanto teste) exige interação com o usuário. Sendo assim, os resumos

foram divididos em dois conjuntos: aproximadamente dois terços para treino (28 resumos) e um terço para testes (15 textos). A distribuição das categorias pode ser vista na Tabela 28.

Tabela 28: Conjuntos de treino e teste usados na comparação com o *Mover*

Categoria	Conjunto de treino	Conjunto de Teste
Contexto	22	18
Lacuna	4	2
Propósito	29	17
Metodologia	34	15
Resultado	120	50
Conclusão	22	14
Total	231	116

O *Mover* sempre aplica seu sentenciador sobre os textos que processa, e em alguns casos as sentenças do *córpus* foram divididas, erroneamente. Foram poucos casos (4 sentenças), e quando ocorreram foram corrigidos para que a comparação pudesse ser feita com o mesmo número de sentenças do *córpus* de teste (116). A correção foi feita da seguinte maneira:

- Se duas ou mais sentenças foram agrupadas, a categoria dada a sentença agrupada é dada as todas as sentenças agrupadas
- Se a sentença foi dividida, analisa as categorias dadas aos segmentos de uma mesma sentença
 - A classe mais freqüente entre os segmentos é dada a sentença
 - Se houver empate, a classe mais freqüente do *córpus* de treino (*Resultado*) é dada à sentença

Como *baseline* foi usado o classificador que rotulou todos as sentenças do conjunto de teste com a categoria mais freqüente no treinamento (*Resultado*). Os resultados da taxa de acerto e estatística *Kappa* podem ser observados na Tabela 29

Tabela 29: Comparação entre sistemas AZEA, *Mover* e *baseline*

	Acerto	Kappa	Macro-F
<i>AZEA</i>	73,2%	0,63	70%
<i>Mover</i>	56,9%	0,27	34%
<i>baseline</i>	43,1%	0	10%

A taxa de acerto do *Mover* foi de 56,9% e o valor do Kappa foi de 0,267, resultados realmente muito baixos. Na Tabela 30 podemos observar que a estratégia do *Mover*

assemelha-se à *baseline*, pois rotulou muitos exemplos com a classe majoritária (coluna **R**). Entretanto, o corpus de treinamento é muito pequeno para o treinamento de métodos puramente estatísticos.

Tabela 30: Matriz de confusão para o *Mover*

	Ct	L	P	M	R	Co
Ct	5	0	0	13	0	0
L	0	0	0	0	2	0
P	0	0	8	1	8	0
M	0	0	0	0	15	0
R	0	0	0	0	47	3
Co	1	0	0	0	7	6

Por último temos as estatísticas detalhadas por categorias, do *Mover* e do AZEA, que são mostradas na Tabela 31. Se compararmos categoria a categoria pela Medida-F, todas as categorias do *AZEA* obtiveram melhores resultados que as do *Mover*. A única métrica cujo *Mover* se sobressaiu foi a precisão da categoria *Propósito*, com 100% do *Mover* contra 62,5% do AZEA. Esse fato aponta que uma composição do *Mover* e o *AZEA* usando a abordagem de *Purpose Splitter* talvez fosse adequada, garantiria a identificação das sentenças de propósito sem cometer erros.

Tabela 31: Estatísticas detalhadas por categoria do *Mover* e do AZEA

Categoria	Prec.	Cob.	F	Prec.	Cob.	F
<i>Contexto</i>	83,3%	27,8%	42%	90%	50%	64,3%
<i>Lacuna</i>	0%	0%	0%	100%	50%	66,7%
<i>Propósito</i>	100%	47%	64%	62,5%	88,2%	73,2%
<i>Metodologia</i>	0%	0%	0%	46,2%	40%	42,9%
<i>Resultado</i>	51,1%	94%	66%	75,5%	80%	77,7%
<i>Conclusão</i>	42,9%	42,9%	52%	86,7%	92,9%	89,7%

6.5 *AZSections*: Estendendo a detecção para outras seções do texto científico

Mostramos nesta seção que alguns atributos do *AZEA* podem ser utilizados como um núcleo básico de detecção da estrutura esquemática, podendo ser aprimorado posteriormente, aproveitando-se das peculiaridades da seção sendo abordada. Propomos com o *AZSections* um núcleo básico de atributos que possa ser usado como um bom ponto de partida para um classificador da estrutura esquemática para as outras seções de um artigo.

6.5.1 Córpus

Foram utilizados cinco córpus, todos extraídos do SciPo-Farmácia: *Conclusões*, *Discussões*, *Introduções*, *Metodologia* e *Resultados*. Na Tabela 32, podemos observar algumas estatísticas de cada córpus. Em cada linha é mostrado um córpus, com as seguintes estatísticas: total de textos, total de sentenças, total de palavras, total de categorias distintas e a porcentagem de sentenças rotuladas com a categoria majoritária.

Tabela 32: Estatísticas dos córpus utilizados no treinamento e avaliação do *AZSections*

Córpus	# textos	# sent.	# pal.	# cat.	% freq.
conclusões	22	173	5385	5	57.8%
discussões	12	314	9141	6	42.3%
introduções	39	812	24304	8	37.5%
metodologias	30	1041	24454	5	72.8%
resultados	26	1429	39370	5	90.5%

Na Tabela 33, são mostradas as categorias de cada um dos córpus. Na segunda coluna está o número de categorias esquemáticas de cada córpus, seguida pela coluna em que seus nomes são citados. Entre parênteses, é dada também a porcentagem de sentenças da categoria no córpus.

Tabela 33: Categorias esquemáticas de cada um dos córpus do SciPo-Farmácia usados no *AZSections*

Córpus	#	Categorias
conclusões	5	Conclusão(24,3%), Contexto(0,6%), Metodologia(6,4%), Propósito(11,0%), Resultado(57,8%)
discussões	6	Apresentar conclusões(8,3%), Relacionar a pesquisa com a literatura(39,5%), Retomar contexto(4,8%), Retomar metodologia(2,9%), Retomar propósito(2,2%), Revisar resultados mais importantes(42,4%)
introduções	8	Contexto(37,6%), Estrutura(0,9%), Justificativa/Valor(0,1%), Lacuna(8,1%), Metodologia(2,3%), Propósito(8,1%), Resultado(8,7%), Revisão da literatura(34,1%)
metodologias	5	Análise dos Dados(7,2%), Equipamentos(3,3%), Materiais(13,6%), Procedimento(72,8%), Resultados(3,1%)
resultados	5	Bibliografia/literatura(2,7%), Contexto(2,0%), Metodologia(3,4%), Propósito(1,3%), Resultado(90,6%)

6.5.2 Atributos

Os atributos utilizados são um subconjunto dos atributos descritos na Seção 6.3. São os mesmos utilizadas no *AZEA*, com exceção daquelas intrinsicamente ligadas à seção de *abstracts*. O conjunto de atributos usado pelo *AZSections*, com oito atributos, é mostrado na Tabela 34, novamente com uma breve descrição.

Tabela 34: Atributos utilizados no <i>AZSections</i>	
Atributo	Descrição
Localização	Posição da sentença no texto
Histórico	Categoria da sentença anterior
Tempo	Tempo verbal do primeiro verbo finito da sentença
Modal	Existência de auxiliar modal
Voz	Voz verbal
<i>Formulaic Expressions</i>	Tipo de expressão-padrão contido na sentença
Agente	Tipo de agente contido na sentença
Tamanho	Comprimento da sentença em palavras

Basicamente, temos todos os grupos de atributos representados: contextuais, sintaxe verbal, tamanho de sentença e expressões-padrão. A extração dos atributos é feita da mesma forma, como anteriormente descrito na Seção 6.3.

6.5.3 Treinamento

Inicialmente, os corpúscos foram pré-processados, e foram geradas tabelas atributo-valor para cada um dos corpúscos. Foram aplicados quatro algoritmos de aprendizado de máquina distintos sobre cada corpúscos, com o WEKA. O método de estimativa de erro real utilizado foi o *10-fold stratified cross-validation*. Os valores de precisão e *Kappa* podem ser observados na Tabela 35.

Como *baseline*, foi utilizado um classificador que sempre rotula uma sentença com o classe majoritária de cada corpúscos no treinamento. Os melhores resultados foram obtidos no corpúscos de introduções (*Kappa* de 0,667 e acerto de 76,23%) e embora o corpúscos de resultados tenha registrado uma taxa de acerto muito alta (90,9%), seu *Kappa* é baixo (0,17). Tal fato deve-se ao desbalanceamento do corpúscos: como vemos na *baseline*, 90,5% das sentenças são da classe majoritária, e para o corpúscos de metodologias o valor de *Kappa* também foi baixo (*Kappa* de 0,38, com 78,4% de acerto com 72,8% da *baseline*).

Tabela 35: Precisão e *Kappa* para os corpúscos do SciPo-Farmácia

<i>Corpus</i>	Naive Bayes	J48	JRip	SMO	Baseline
conclusões	72,8% 0,48	73,4% 0,50	75,1% 0,54	70,5% 0,46	57,8% 0
discussões	63,3% 0,39	64,6% 0,43	63% 0,41	62,1% 0,39	42,3% 0
introduções	76,2% 0,66	75,3% 0,65	75,3% 0,64	76,2% 0,667	37,5% 0
metodologias	78,4% 0,38	76,6% 0,34	76,1% 0,34	76,6% 0,37	72,8% 0
resultados	90,9% 0,17	90,5% 0,09	90,2% 0,08	90,6% 0,14	90,5% 0

6.6 Considerações finais

Os resultados obtidos pelo *AZEA*⁸ figuram entre os resultados relatados na literatura. No conjunto de treinamento, o *Kappa* (K) do *AZEA* foi de $K = 0,73$ com o classificador induzido com o algoritmo SMO. Diferentes algoritmos obtiveram diferentes resultados sobre distintos corpúscos de teste, e o ranqueamento variou em grande parte dos diferentes testes. O J48 obteve os melhores quando aplicado aos resumos de alunos ($K = 0,73$), e o *Naive Bayes* obteve o maior desempenho entre os resumos atípicos ($K = 0,55$). O resultado no corpúscos *Atípicos* (*Kappa* 30% inferior ao obtido no treinamento) mostra que o *AZEA* é muito sensível e dependente da estrutura esquemática do resumo a ser classificado.

Embora tenhamos testado o nosso método para diferentes seções, não temos nenhuma estimativa de como o *AZEA* se comporta ao receber resumos de outras áreas. Contudo, nossa metodologia requer menos recursos manualmente anotados para atingir bons resultados. Quando confrontado com o *Mover*, apesar de apenas 28 resumos no conjunto de treino, o *AZEA* obteve um bom *Kappa* ($K = 0,63$), enquanto que o *Mover* registrou $K = 0,27$ e sua classificação assemelhou-se muito a praticada pela *baseline*.

O *AZEA* pode ser utilizado também para auxiliar a tarefa de anotação manual da estrutura esquemática, como um segundo anotador, visto que o *Kappa* obtido ($K = 0,73$) é muito próximo ao relatado entre dois humanos no *AZPort* ($K = 0,69$) (FELTRIM, 2004), cujo esquema de anotação é muito semelhante. Além disso, a discordância feita pelo *AZEA* aparenta estar isolada em três pares de categorias: *Metodologia X Resultado*, *Resultado X Conclusão* e *Contexto X Propósito*. Primeiramente, precisaríamos aferir a concordância entre anotadores humanos sobre esses pares de categorias. Uma outra solução seria alterar

⁸O *AZEA* está disponível para testes no endereço <http://www.nilc.icmc.usp.br/azea-web/>

a unidade de classificação, classificando orações em vez de sentenças, embora os erros para a segmentação de sentenças em orações provavelmente se sobressairiam sobre os problemas de confusão entre categorias.

A proposta de estender o método para outras seções, o *AZSections*, obteve resultados regulares, com exceção do bom resultado da seção de *introdução* ($K = 0,66$) e resultado ruim da seção *resultados* ($K = 0,17$). Em parte, isso deve-se a inflexibilidade dos limiares utilizados na extração dos atributos. No resumo, faz sentido o atributo *Localização* diferenciar somente as duas primeiras, as duas últimas e as sentenças medianas, pois em média os resumos têm 8 sentenças. Já na seção *resultados*, cada texto tem 60 sentenças em média, e 56 (93%) dessas sentenças receberão o mesmo valor de atributo.

Para a melhoria do *AZSections*, dois pontos devem ser trabalhados. Primeiramente, o ajuste, no próprio corpus de treinamento, dos limiares dos atributos, principalmente os atributos *Localização* e *Comprimento de sentença*. Em segundo, a extração de expressões-padrão do próprio corpus de treinamento. Novos métodos automáticos têm sido desenvolvidos para a extração de expressões-padrão (ABDALLA; TEUFEL, 2006). Este processo, quando feito manualmente, demanda muito tempo, mesmo de lingüistas e especialistas em gênero científico.

7 *Implementando as dimensões de uma rubrica baseada no gênero científico*

Neste capítulo, descrevemos algumas das incursões na avaliação da qualidade de escrita de resumos científicos escritos em inglês. Focamos nosso trabalho na automatização das dimensões D1, D2 e parte da dimensão D5 da rubrica descrita no Capítulo 5.

7.1 Implementações das dimensões relacionadas com organização e balanceamento

Embora a taxa de acerto obtida pelo nosso classificador da estrutura esquemática seja de 80%, o valor de *Kappa* foi substancial (0,73), e acreditamos que ele possa ser usado para identificar a estrutura esquemática, possibilitando que as dimensões da rubrica que baseiam-se em informações da EE seja implementadas. Duas das sete dimensões, D1 e D2, são facilmente implementadas quando a EE do resumo está acessível.

As duas dimensões fornecem, juntas, 8 mensagens de *feedback*. As mensagens são bem claras e imediatas, com exceção da mensagem “Problemas na organização de sentenças”, em que o usuário necessita conhecer a ordem esperada das componentes da estrutura esquemática para efetuar correções.

- **D1: Presença e organização de componentes**

- Problemas na organização de sentenças
- Falta componente: *purpose*
- Falta componente: *result*
- Falta componente: *conclusion*

- Componente 'gap' existe, mas não está seguida pelo propósito

- **D2: Balanceamento entre componentes**

- Componente 'purpose' deve existir, escrita em uma única sentença
- Componente 'conclusion' deve existir, escrita em uma única sentença
- Componente 'background' excede, em número de palavras, 30% do resumo

No protótipo que desenvolvemos para a Web, o *AZEA-Web*¹, ao final do texto anotado com a estrutura esquemática são reportadas as críticas baseadas nas dimensões D1 e D2 da rubrica. Um resumo submetido ao *AZEA-Web* teve sua estrutura esquemática detectada, e na Figura 29 são mostradas as críticas baseadas nas dimensões D1 e D2 à estrutura detectada. Os erros são apontados para que o usuário possa melhorar a estrutura do seu resumo. Também são dadas algumas estatísticas sobre a distribuição das componentes no resumo, dados estes que ajudam na correção de erros na dimensão D2, principalmente.

Figura 29: Críticas das dimensões D1 e D2 fornecidas pelo *AZEA-Web*

¹<http://www.nilc.icmc.usp.br/azea-web>

7.2 Implementação de um detector automático de erros de uso de artigos

Como descrito na Seção 5.4, a dimensão D5 engloba diversos erros técnicos cometidos por estudantes. Exemplos de erros são o uso incorreto de palavras, a confusão na escolha de preposições, erros de emprego de tempo verbal, erros de ortografia, entre outros. Dentre os mais freqüentes está o erro de uso do artigo, que reside em decidir se um determinado **sintagma nominal** (SN) deve conter um dos artigos da língua inglesa (*the*, *a* ou *an*) precedendo o núcleo do sintagma nominal.

Decidimos implementar um detector automático de erro de uso de artigo em inglês por vários motivos. Primeiro, é um dos erros mais representativos da dimensão D5 da rubrica e um dos mais cometidos pelos estudantes brasileiros. Além disso, é uma tarefa bem definida e explorada na literatura (uma boa revisão destes trabalhos pode ser encontrada em (HAN et al., 2006)). Acreditamos também que os resultados deste experimento podem ajudar a estimar o tempo e os recursos gastos para implementar outras dimensões da rubrica ou mesmo outros detectores para outros tipos de erros da própria dimensão D5.

Baseamos algumas decisões de implementação do nosso detector de erros de uso de artigos nos experimentos realizados para a definição da rubrica, descrita na Seção 5.4. Dos 23 tipos possíveis de erros marcados (mostrados no Apêndice B), poucos deles foram responsáveis pela grande maioria dos erros cometidos no geral. O erro de uso de artigo, que foi dividido em três tipos, foi responsável por 19,2% do total de erros cometidos no corpus. Os três tipos de erros são descritos a seguir:

ART- : O sintagma nominal **não** contém um artigo, mas deveria;

ART+ : O sintagma nominal contém um artigo, mas **não** deveria;

ART : O sintagma nominal contém um artigo, mas o artigo errado (*the* em vez de *a/an*, e vice-versa).

Dentre os erros de uso de artigo, o mais freqüente é o **ART-** (70%), seguido de **ART+** (26%) e **ART** (4%). Podemos aferir pela distribuição dos erros de uso de artigo que o problema consiste em determinar quando um artigo deve ser usado ou não.

7.2.1 Trabalhos relacionados sobre a detecção automática de erros de uso de artigo em inglês

Grande parte dos estudos de detecção automática de erros de uso de artigo em inglês são baseados ou em construção de regras ou no uso de métodos estatísticos sobre grandes corpú. Han et al. (2006) fazem uma ótima descrição de trabalhos com essas diferentes abordagens. Grande parte dos trabalhos contextualiza a determinação automática do artigo em sistemas de tradução da língua japonesa para a língua inglesa².

Atualmente, um modelo bastante utilizado é o estatístico de máxima entropia. As abordagens baseadas neste método têm obtido bons resultados (HAN et al., 2006). Entretanto, sistemas baseados em máxima entropia necessitam de grandes corpú para seu treinamento. Lee (2004) relata que usou um corpú com cerca de 260.000 SNs, e Han et al. (2006) treinou seu modelo sobre cerca de 8 milhões de SNs. Para métodos baseados em aprendizado de máquina, ressaltamos o trabalho pioneiro de Knight & Chander (1994), que propõe seu método como um pós-processamento de traduções para o inglês, selecionando com 78% de precisão a escolha entre artigos definidos e indefinidos.

Nos baseamos no trabalho de Han et al. (2006), que propõe 11 combinações de diferentes atributos para a detecção de três possíveis usos do artigo, *null*³, *the* ou *a/an*. Tentamos mapear grande parte dos tipos de atributos utilizados e quais *tokens* são utilizados para o desenvolvimento de nossos atributos. O contexto do qual são extraídos os atributos é composto por dois *tokens* que precedem o SN (*pre-pre-NP* e *pre-NP*), os *tokens* do SN e um *token* após o SN (*pos-NP*).

Os tipos de atributos extraídos e utilizados são os próprios *tokens*, as etiquetas morfosintáticas desses *tokens* e um atributo que tenta decidir se o núcleo do SN é contável ou não. Este atributo, denominado *Countability*, é determinado com o auxílio de medidas de frequência extraídas do corpú de treinamento. A Tabela 36 resume os possíveis valores utilizados na extração do atributo *Countability*, e a condição para que recebam tais valores.

Os resultados relatados na literatura estão em torno de 80% de acerto. Han et al. (2006) obteve 83% de acerto em textos publicados e, na comparação com anotadores humanos, concordaram em 85% dos casos (*Kappa* de 0,48). Minnen et al. (2000) relata 82,6% de acerto as classes *null*, *the* e *a/an*.

²É importante ressaltar que o japonês, assim como o russo e algumas outras línguas, não possui artigos.

³Neste capítulo será utilizado o termo *null* para designar a ausência do artigo em frente a um sintagma nominal.

Valor	Condição
<i>uncountable</i>	Se a ocorrência do núcleo no plural for menor de 3% de sua ocorrência total
<i>pluralia tantum</i>	Se a ocorrência do núcleo no plural for superior a 95% de sua ocorrência total
<i>unknown</i>	se o núcleo não for encontrado no <i>corpus</i> de treinamento
<i>countable</i>	caso contrário

Tabela 36: Valores do atributo *Countability* no trabalho de (HAN et al., 2006).

7.3 Detecção automática de erros de uso de artigos em resumos científico em inglês

Amparados pelos trabalhos correlatos que citamos na Seção 7.2.1, construímos um protótipo de detector automático de erros de uso de artigos. Fundamentamos algumas decisões baseadas em dois pontos principais, sendo eles o gênero textual e o público-alvo do *feedback* deste erro. Nossa abordagem é baseada no estudo de textos do gênero científico, enquanto que os trabalhos anteriores utilizaram um grande volume de textos jornalísticos e dissertações de tema livre (*essays*), tanto no treinamento quanto em seus testes. Além disso, grande parte dos estudos foca em erros cometidos por falantes nativos de línguas que não têm artigos, como o japonês, enquanto nosso público-alvo são os estudantes brasileiros, cujos erros de uso de artigo já sabemos que concentram-se na colocação ou não do artigo.

Essas decisões afetaram profundamente o projeto do detector automático de erros de uso de artigos em inglês que apresentamos nesta seção. Primeiramente, precisávamos montar um *corpus*. Decidimos não utilizar *corpus* jornalísticos, e construímos manualmente nosso *corpus*, descrito na Seção 7.3.1, que contém apenas resumos científicos.

Não usamos o método baseado em máxima entropia para o treinamento. O motivo está no fato que nosso *corpus* é imensamente menor que os *corpus* utilizados nos trabalhos em que a máxima entropia é empregada. Contudo, buscamos extrair os atributos mais utilizados nos trabalhos correlatos, principalmente os atributos descritos em (HAN et al., 2006) e um atributo descrito em (LEE, 2004). Houve ajustes para que pudéssemos modelar os atributos flexíveis para nosso conjunto fixo de atributos, como o estabelecimento de um número máximo de *tokens* considerados de cada sintagma nominal.

Utilizamos a mesma metodologia aplicada no desenvolvimento dos experimentos com a detecção automática da estrutura esquemática, e aplicamos vários indutores de diferentes algoritmos do ambiente WEKA sobre os valores de atributos que extraímos.

Por fim, nos aproveitamos dos dados obtidos no estudo dos erros de estudantes brasileiros, que nos permitiram fazer algumas asserções que não parecem cabíveis se tratarmos outro público-alvo, com a perspectiva de um futuro acoplamento deste detector à sistemas de auxílio de escrita científica.

7.3.1 Córpus

Nosso córpus é composto de 723 resumos científicos, das áreas de Farmácia (354) e Física (369). Estes resumos foram obtidos na Internet, com exceção de 74 resumos da área de Farmácia que foram incorporados do córpus de treinamento do *AZEA*, descrito na Seção 3.2. Os outros resumos foram retirados de grandes revistas, tais como *Physical Review Letters (A-D)*, *Science*, *Nature*, *Biotechnology Progress*, entre outras. As estatísticas lexicais deste córpus são mostradas na Tabela 37.

Tabela 37: Estatísticas de cada texto do córpus

	Total	Média	Desv. Padrão
Sentenças	4886	6,76	(2,78)
Palavras	115913	160,3	(67,04)
SN	31960	44,2	(19,31)

Cada um dos textos do córpus foi sentenciado, tokenizado e segmentado em *chunks*. Tanto o sentenciador quanto o *chunker* usados foram os disponíveis no pacote *OpenNLP*, enquanto que o tokenizador utilizado foi o criado pelo *Penn Treebank*, todos descritos na Seção 3.3. Após este processo, foram isolados apenas os SNs, totalizando 31960 SNs, sendo que cada um tem, em média, $2,41 (\pm 1,25)$ *tokens*.

Cada um dos SN pertence a uma das três classes: *the*, *a/an* ou *null*. A classe mais numerosa do córpus é a *null* (65,7%), seguido por *the* (25,3%) e *a/an* (9%).

Também é importante avaliarmos como o nosso detector, treinado com resumos publicados, comporta-se ao avaliar resumos elaborados por estudantes. Para tanto, utilizamos um subcórpus dos textos produzidos pelos estudantes brasileiros, que foi analisado por um falante nativo de inglês (JR. et al.,). Este córpus é composto por 78 resumos, de três cursos de escrita científica em inglês para brasileiros, estudantes de pós-graduação das áreas de Ciências Farmacêuticas e Física. Das 570 sentenças desse córpus, foram extraídos 3585 SNs, sendo que cada sentença tem $6,2 (\pm 3,4)$ SNs em média. Este córpus está em formato DOC, e foi convertido primeiramente para HTML, com auxílio da ferramenta de

conversão *Abiword*, e depois para texto plano⁴ (via ferramenta *Lynx*). Os erros encontrados no resumo estão marcados com etiquetas, que foram separadas e são usadas posteriormente neste capítulo para avaliação de nosso detector sobre os erros realmente cometidos pelos alunos.

7.3.2 Atributos

Cada SN é representado pelos valores de 39 atributos. Descrevemos primeiramente os *tokens* envolvidos na extração dos valores dos atributos, que denominamos de *contexto local*. Depois descrevemos os seis tipos de atributos que podem ser aplicados sobre estes *tokens*. Cada atributo, portanto, é um tipo de atributo aplicado à um *token* do contexto local.

7.3.2.1 Contexto local

Os atributos são extraídos de uma sequência de *tokens* em torno do SN a ser classificado, como proposto em (HAN et al., 2006), que chamamos de *contexto local*. O contexto local é composto de três regiões:

1. **SN *tokens*:** São os *tokens* contidos no SN. Como utilizamos um conjunto fixo de atributos, definimos uma janela de *tokens* que são considerados, para delimitar o máximo de *tokens* que serão analisados. Esta janela é composta por até oito *tokens*, sendo quatro anteriores e quatro posteriores ao núcleo do SN. O primeiro *token* (*Inicial*) do SN é uma posição considerada à parte, embora possa pertencer à janela também. Se este *token* for o artigo, o próximo *token* (à direita) do SN representa esta posição. No total, são nove posições de *tokens* que são consideradas. As posições da janela seguem a ordem descrita à seguir (o núcleo não faz parte da janela; é o *token* entre W_{B1} e W_{F1}):

Janela: $\{W_{B4}, W_{B3}, W_{B2}, W_{B1}, * * head * *, W_{F1}, W_{F2}, W_{F3}, W_{F4}\}$

2. **Núcleo do SN ou *head*:** O núcleo do sintagma nominal. O núcleo é definido por regras descritas em (COLLINS, 1999). Embora faça parte do SN, o definimos como uma posição separada, pois vários atributos são extraídos dele.
3. ***Tokens* periféricos:** São os *tokens* ao redor do SN. São considerados os dois *tokens* antes do SN (*pre-pre-NP* e *pre-NP*) e um *token* após (*pos-NP*). Os limites sentenciais não são ultrapassados.

⁴Estas ferramentas estão descritas na Seção 3.3.

A Figura 30 ilustra o contexto local explorado sobre um trecho do *cópus*, já pré-processado. As etiquetas morfossintáticas podem ser vista no Anexo A.

Figura 30: Contexto considerado na extração dos atributos

7.3.2.2 Tipos de atributo

Abaixo são descritos os 6 tipos de atributos que podem ser aplicados aos *tokens* do contexto local. O número de atributos gerados pelos tipos descritos varia desde apenas 1 até 13. Metade dos tipos de atributos precisa ser “calibrada” previamente no *cópus* de treinamento, como os atributos *Palavras*, *Rice of Party* e *Countability*. O tipo de atributo *Discurso* mantém uma lista dinâmica que é atualizada no decorrer do processamento de um novo texto. Os demais tipos, se já calibrados, utilizam recursos estáticos sobre os *tokens* do contexto local.

1. **Artigo:** é o atributo que representa a classe. Os *tokens* do SN anteriores ao núcleo do SN são analisados, à procura de um artigo. Pode assumir três valores: *null*, se nenhum artigo for encontrado; *the*, se o artigo definido for encontrado ou *a/an*, se o artigo encontrado for indefinido.
2. **Etiquetas:** As etiquetas morfossintáticas de todos os *tokens* que estão no contexto local. As etiquetas são as mesmas utilizadas na anotação do *Penn Treebank*, e podem ser vistas no Anexo A. Se o *token* não puder ser determinado, o atributo recebe *unknown*. Por exemplo, nos casos em que o núcleo é o último *token* do SN, os quatro *tokens* posteriores ao núcleo não podem ser determinados.
3. **Palavras:** confronta o *token* com uma lista de palavras, composta pelas 35 palavras mais frequentes do *cópus*, com exceção dos artigos⁵. Os valores deste tipo de atributo são uma das 35 palavras da lista mostrada na Figura 31 ou *unknown*, se *token* não

⁵Este limiar de 35 foi definido experimentalmente no *cópus*. Testes com mais de 35 palavras ou remoção de *stopwords* não apresentaram melhora no desempenho do classificador. Os artigos foram excluídos da lista pois supõe-se que nenhum dos *tokens* analisados seja um artigo.

for encontrado na lista. É extraído o valor deste atributo de cada um dos 13 *tokens* do contexto local. Se alguns dos *tokens* do contexto local não puder ser determinado, cada um deles recebe o valor *unknown*.

Figura 31: 35 palavras mais freqüentes utilizadas como valores do tipo de atributo *Palavra*. O token “t” representa um *not* contraído (como em *don’t*)

of	that	this	these	can
and	by	from	cell	have
in	we	cells	or	also
to	was	be	using	model
for	are	were	t	results
with	on	at	protein	expression
is	as	which	not	it

4. ***Rice of Party***: busca se o *token* é sempre visto em um SN de uma determinada classe. Antes do processo de extração dos valores dos atributos, é calculada a lista de todas as palavras contidas nos SN do córpis com freqüência superior ou igual a 5, separadas pela classe do SN em que foram encontradas. Somente as palavras que aparecem apenas em uma lista são consideradas. Este tipo de atributo é aplicado a 10 atributos, sendo eles os *SN tokens* e ao núcleo do SN. Os valores deste tipo de atributo são os próprios valores de *Artigo* mais *unknown*, atribuído quando a palavra não está em nenhuma das listas.
5. ***Countability***: define se o núcleo do SN é contável. Para extrair este atributo, nos baseamos no trabalho de Han et al. (2006). Primeiramente, o córpis é processado e é gerada uma lista de freqüências, com todas as palavras do córpis. Depois, para cada palavra w_i da lista busca-se seu par em número, que se não encontrado recebe freqüência 0. Com os valores das freqüências dos termos w_i no singular ($f_{SG}(w_i)$) e no plural ($f_{PL}(w_i)$), podemos gerar uma lista de *Countability*, e o termo w_i será armazenado com um dos valores do atributo, que é calculado como foi descrito anteriormente na Tabela 36. Para determinar o número da palavra, foi utilizado o pacote Perl *Lingua-EN-Inflect-Number-1.1*⁶. Como nosso córpis é muito pequeno, a lista de freqüência do córpis BNC⁷ foi unida com a nossa lista de freqüência. Foi aplicado um corte na lista de freqüência do BNC, e palavras com ocorrência igual ou inferior a 10 foram desconsideradas. A lista resultante chamamos de *countability list*. Ao extrair este atributo de um novo SN, o núcleo é buscado na *countability list*. Se for

⁶<http://search.cpan.org/dist/Lingua-EN-Inflect-Number/>

⁷<http://www.natcorp.ox.ac.uk/>

encontrado, retorna o valor de *Countability* (*countable*, *uncountable* ou *pluralia tantum*) armazenado juntamente com a palavra. O atributo recebe *unknown* se o núcleo não estiver na lista.

6. **Discurso:** indica se o núcleo do SN já foi citado anteriormente ou se é a primeira vez que aparece no resumo. As palavras são armazenadas à medida que o texto vai sendo processado. Se o núcleo do SN sendo processado no momento já estiver na lista, então o valor deste atributo é *seen*; caso contrário, recebe *new*. Lee (2004) sugere que se restrinja a busca às cinco sentenças anteriores ao SN sendo avaliado. Em nosso trabalho não aplicamos nenhuma restrição por trabalharmos com resumos, mas isto pode ser necessário para textos maiores.

Os tipos de atributos são mostrados na Tabela 38, juntamente com sua descrição e mais informações do número de atributos, a que *tokens* são aplicados e seus valores.

Tabela 38: Os 39 atributos, separados por grupos, utilizados na classificação

Tipos de atributos	<i>Tokens</i>	# atrib.	Descrição	Valores
<i>Artigo</i>	–	1	Artigo que encabeça o SN (classe)	<i>the</i> , <i>a/an</i> , <i>null</i>
Etiquetas	contexto local	13	Etiqueta morfosintática do <i>tokens</i>	Etiquetas do <i>Penn Treebank</i>
Palavras	contexto local	13	Se o <i>token</i> é uma das 35 palavras mais freqüentes do texto	Uma das 35 palavras ou <i>unknown</i>
<i>Rice of Party</i>	SN Tokens, núcleo SN	10	<i>token</i> comumente acompanha SNs de uma das três classes	<i>the</i> , <i>a/an</i> , <i>null</i> ou <i>unknown</i>
<i>Countability</i>	núcleo SN	1	Se o núcleo do SN é contável	<i>countable</i> , <i>pluralia tantum</i> , <i>uncountable</i> ou <i>unknown</i>
Discurso	núcleo SN	1	Se o núcleo do SN já apareceu no texto	<i>new</i> ou <i>unseen</i>

Como exemplo, extraímos os atributos do SN “*the cell membrane*” mostrado na Figura 30. Primeiramente, temos que vários *tokens* dos *SN Tokens* não podem ser determinados. De fato, apenas dois *tokens* podem ser determinados: *Inicial* e W_{B1} . A extração de cada um dos tipos de atributos é ilustrada na Figura 32.

1. **Artigo:** *the*
2. **Etiquetas:** $W_{B4} = W_{B3} = W_{B2} = W_{F1} = W_{F2} = W_{F3} = W_{F4} = \text{unknown}$
 $W_{B1} = \text{NN}$
 $\text{Inicial} = \text{NN}$
 $\text{pre-pre-NP} = \text{WDT}$, $\text{pre-NP} = \text{VBP}$ e $\text{pos-NP} = \text{IN}$
 $\text{head} = \text{NN}$
3. **Palavras:** $W_{B4} = W_{B3} = W_{B2} = W_{F1} = W_{F2} = W_{F3} = W_{F4} = \text{unknown}$
 $W_{B1} = \text{cell}$
 $\text{Inicial} = \text{cell}$
 $\text{pre-pre-NP} = \text{that}$, $\text{pre-NP} = \text{unknown}$ e $\text{pos-NP} = \text{in}$
 $\text{head} = \text{unknown}$
4. **Rice of Party:** $W_{B4} = W_{B3} = W_{B2} = W_{F1} = W_{F2} = W_{F3} = W_{F4} = \text{unknown}$
 $W_{B1} = \text{unknown}$
 $\text{Inicial} = \text{unknown}$ $\text{head} = \text{unknown}$
5. **Countability:** $\text{head} = \text{countable}$
6. **Discurso:** $\text{head} = \text{new}$

Figura 32: Exemplo de extração dos valores dos atributos para o detector automático de erros de uso de artigo em inglês. Para simplificar, tratamos o trecho em questão como se fosse o texto sendo processado, e por isso o valor *new* do atributo *Discurso*

7.3.3 Treinamento e avaliação

Como já adiantamos, a método mais utilizado para a detecção automática de erros de uso de artigos em inglês é a máxima entropia (RATNAPARKHI, 1997) (veja (HAN et al., 2006) e (LEE, 2004)). Contudo, classificadores baseados em máxima entropia necessitam de grandes corpú de treinamento, e preferimos utilizar o ambiente WEKA. Nele ainda pudemos testar diferentes algoritmos, comumente empregados em tarefas de Categorização Textual e *Text Mining*.

Os algoritmos utilizados foram: **J48**, **JRip** e **Naive Bayes (NB)**. O método usado para a estimativa do erro real foi o *10 fold cross-validation*. Usamos a taxa de acerto e a medida *Kappa* para a avaliação, assim como as medidas precisão, cobertura e medida-F. A *baseline*, sempre que referenciada, é o classificador que rotula um novo caso sempre com a classe majoritária no treinamento (*null*, em nosso corpú).

Na Tabela 39, são descritos os valores de taxa de acerto e medida *Kappa* dos classificadores induzidos e da *baseline*.

Podemos observar que os valores de *Kappa* são baixos, entre 0,34 e 0,50. Os valores de

Tabela 39: Valores da taxa de acerto e $Kappa$ sobre as três possíveis classes de artigos

Classificador	Acerto	$Kappa$
J48	77,4%	0,50
NB	70,2%	0,44
JRip	73,8%	0,34
<i>baseline</i>	65,7%	0

precisão, cobertura e medida-F em de cada uma das três classes são mostrados na Tabela 40.

Tabela 40: Precisão (P), Cobertura (C) e medida-F (F) para os classificadores sobre as três classes

	J48			NB			JRip		
	P	C	F	P	C	F	P	C	F
<i>null</i>	83%	93%	88%	87%	76%	81%	75%	97%	84%
<i>the</i>	64%	55%	59%	49%	65%	56%	70%	35%	47%
<i>a/an</i>	49%	28%	35%	44%	44%	44%	63%	10%	17%

Podemos observar que a medida-F respeita a ordem de distribuição de casos por classes. A classe *null* obteve valores de medida-F superiores a 80%, enquanto que a classe *a/an* teve resultados muito ruins, abaixo de 45%.

Outra questão que levantamos é a dependência do classificador à área dos resumos do treinamento. Para esta avaliação, separamos o nosso corpus de treinamento por suas áreas: Física, com 369 resumos (12427 SNs) e Farmácia, com 354 resumos (19533 SNs). Notamos que o valor de $Kappa$ para ambos os corpus de resumos se manteve, apesar de uma ligeira alteração nos valores da taxa de acerto. Esses resultados são mostrados na Tabela 41, onde as linhas são os corpus de teste e as colunas são os corpus de treinamento. Ou seja, na linha *Farmácia* e coluna *Física* são mostrados os valores da taxa de acerto e $Kappa$ (entre parênteses) para o classificador induzido sobre o corpus de resumos de Física e testado sobre o corpus de Farmácia. As posições da matriz cuja linha é igual a coluna reportam valores obtidos via *10-fold cross-validation*. O indutor utilizado foi o J48.

Tabela 41: Classificadores treinados com o indutor J48 sobre corpus com três classes, separados por área (Farmácia e Física). São reportados os valores de acerto e o valor de $Kappa$ entre parênteses

	Farmácia	Física
Farmácia	80,6% (0,48)	76,9% (0,48)
Física	73,5% (0,51)	69,5% (0,40)

O corpus de Farmácia, quando rotulado pelo classificador induzido sobre o corpus de Física, teve uma ligeira queda em relação aos valores obtidos no treinamento do corpus

completo. Entretanto, a aplicação contrária (Farmácia sobre Física) infligiu uma diminuição mais substancial (*Kappa* caiu de 0,55 para 0,40).

7.3.3.1 Experimentos com os classificadores binários *HasArticle* e *DetArticle*

Os erros cometidos por estudantes brasileiros concentram-se em **ART+** e **ART-**, ou seja, o estudante tem maior dificuldade apenas de decidir se insere ou não um artigo. Utilizamos este fato a nosso favor, e dividimos a tarefa de classificação, originalmente com três classes, em duas tarefas binárias, que denominamos *HasArticle* (**HA**) e *DetArticle* (**DA**). Primeiramente, o sintagma nominal é avaliado pelo HA, que indica se este necessita ou não de um artigo. Após essa avaliação, em grande parte dos casos o estudante pode ser capaz de definir sozinho qual artigo (*the* ou *a/an*) deve ser empregado. Se ainda sim o estudante tiver dúvidas, então é acionado o DA, que sugere qual artigo deve ser empregado. Esta abordagem privilegia o *feedback* dado ao aluno, entretanto, os resultados do uso de dois classificadores binários combinados não superou o classificador com todas as três classes.

Para o treinamento do HA, primeiramente foram agrupadas as classes *the* e *a/an*. Conseqüentemente, todos os casos dessas classes no cópús de treino tiveram seus valores do atributo *Artigo* alterado para *article*. O cópús ficou então dividido em *null* (65,7% dos casos) e *article* (34,3% dos casos).

Para o DA, apenas foram excluídos os SNs cujo valor do atributo *Artigo* fosse *null*. A distribuição de casos por classe no cópús foi de 73,8% de casos da classe *the* contra 26,2% da classe *a/an*.

Na Tabela 42 são mostrados os valores das taxas de acerto e *Kappa* para os classificadores HA e DA, para os algoritmos testados. Podemos observar, por meio dos valores do *Kappa* que a tarefa de determinar qual o artigo deve ser usado, feita pelo DA com *Kappa* máximo de 0,39, é mais árdua que a tarefa de decidir se o artigo deve ou não ser posto (maior *Kappa* de 0,63). Isto favorece a nossa abordagem, visto que a função exercida pelo DA não é de grande valia para estudantes brasileiros. Em contrapartida, o *Kappa* obtido no HA (0,63) é bom, além de sua taxa de acerto e *Kappa* superarem os valores obtidos no treinamento com três classes (taxa de acerto de 83,7% contra 77,4% e *Kappa* de 0,63 contra 0,50).

Os valores de precisão, cobertura e medida-F das classes, tanto do HA quanto do DA, são mostradas na Tabela 43.

Para tentar avaliar a contribuição de cada um dos 39 atributos extraídos de cada

Tabela 42: Taxa de acerto e Kappa para os classificadores binários HA e DA

	HA		DA	
	Acerto	<i>Kappa</i>	Acerto	<i>Kappa</i>
J48	83,7%	0,63	78,1%	0,31
NB	78,6%	0,49	78,3%	0,39
JRip	82,5%	0,60	75,9%	0,21
SVM	75,3%	0,40	75%	0,18

Tabela 43: Precisão, cobertura e medida-F para os classificadores HA (J48) e DA (NB)

	HA (J48)		DA (NB)	
	<i>none</i>	<i>article</i>	<i>the</i>	<i>a/an</i>
Precisão	86%	78%	82%	61%
Cobertura	89%	73%	89%	47%
Medida-F	88%	75%	86%	53%

SN, mostramos a taxa de acerto e o valor de *Kappa* para classificadores induzidos em dois momentos: com o atributo omitido (Excluído) e somente com o atributo (Sozinho). Na Tabela 44 são reportados esses valores para o HA e o DA, induzidos com o J48 e o NB, respectivamente.

Tabela 44: Contribuição dos conjuntos de atributos usados pelos classificadores binários

	HA				DA			
	Excluído		Sozinho		Excluído		Sozinho	
	Acerto	<i>Kappa</i>	Acerto	<i>Kappa</i>	Acerto	<i>Kappa</i>	Acerto	<i>Kappa</i>
Etiquetas	75,4%	0,41	76,1%	0,48	75,7%	0,22	75,9%	0,33
Palavras	83,6%	0,63	71,7%	0,30	76,7%	0,36	75,3%	0,19
<i>Rice of Party</i>	82,8%	0,61	67,2%	0,06	78,1%	0,41	74,0%	0,01
Discurso	82,2%	0,60	65,7%	0,00	78,0%	0,40	73,8%	0,00
<i>Countability</i>	80,5%	0,56	65,7%	0,00	77,9%	0,39	73,8%	0,00
Atributos do <i>head</i>	74,6%	0,40	75,7%	0,42	76,2%	0,34	74,1%	0,06

Quando combinamos os dois classificadores HA (J48) e DA (NB) e confrontamos com os classificadores ternários, não obtemos melhoras significativas. Este classificador combinado atingiu a taxa de acerto de 77,5% e o valor do *Kappa* de 0,53, enquanto que o melhor classificador ternário obteve a taxa de 77,4% e *Kappa* de 0,50.

Aplicando ao corpus de textos escritos por alunos

Na Tabela 45 são reportados os resultados dos testes de aplicação dos classificadores HA (J48) e DA (NB), induzidos no corpus de treinamento, sobre o corpus de resumos escritos por alunos.

Tabela 45: Taxa de acerto e *Kappa* para os classificadores HA (J48) e DA (NB) quando aplicados ao corpus de resumos de alunos.

	HA		DA	
	Acerto	<i>Kappa</i>	Acerto	<i>Kappa</i>
J48	81,4%	0,57	81,4%	0,35
NB	73,6%	0,44	80,6%	0,41
JRip	81%	0,55	79,8%	0,35
<i>baseline</i>	68%	0	78%	0

Os resultados obtidos nos resumos dos alunos são muito similares aos estimados no corpus de treinamento. Os indutores J48 para o HA e NB para DA mantiveram-se os melhores classificadores também nesse corpus. Para o DA, embora o J48 tenha a taxa de acerto quase um ponto percentual acima do NB, este (NB) obteve *Kappa* de 0,41, contra 0,35 do J48. Na Tabela 46 podem ser observados os valores das métricas de precisão, cobertura e medida-F para os classificadores binários. As classes majoritárias (*none* e *the*) de ambos têm nitidamente melhores resultados que a outra classe.

Tabela 46: Precisão, cobertura e medida-F para os classificadores HA (J48) e DA (NB) sobre os resumos de estudantes

	HA (J48)		DA (NB)	
	<i>none</i>	<i>article</i>	<i>a/an</i>	<i>the</i>
Precisão	85%	72%	56%	86%
Cobertura	87%	68%	50%	89%
Medida-F	86%	70%	53%	88%

Quando os classificadores binários são combinados, a taxa de acerto foi de 76,3%, e *Kappa* de 0,51. Este valores são bem similares ao valores obtidos se o classificador ternário for aplicado, cuja taxa de acerto é de 78,1% e o *Kappa* é de 0,50.

Como explicado anteriormente, os resumos dos estudantes de nosso corpus foram corrigidos por um especialista fluente em inglês. Sendo assim, isolamos somente os SNs do nosso corpus que continham erros de uso de artigo (ART+ e ART-) marcados. No total, 194 SNs foram separados. Este conjunto de SNs foi classificado então pelo HA (J48), e houve uma brusca queda dos resultados, sendo que a taxa de acerto cai para 53% e o *Kappa* foi 0,08. A grande maioria dos erros cometidos pelo classificador (81%, 74 de 91 erros) foram nos SNs que não estavam encabeçados por artigo, embora precisassem.

7.4 Um classificador híbrido com dimensões da rubrica e técnicas de avaliação automática de qualidade de escrita: experimentos iniciais

Conduzimos experimentos iniciais para avaliar a tarefa de detecção automática de resumos de boa qualidade, utilizando duas categorias: “bom” e “ruim”. O detector será utilizado na criação de córpus de referência para as ferramentas de suporte à escrita como o SciPo-Farmácia. Foram utilizados quatro córpus para representar estas duas categorias, mostrados na Tabela 47.

Tabela 47: Córpus adaptados para os testes de detecção de resumos de boa qualidade

Categoria	Córpus	# textos
“Bom”	Treino do AZEA (<i>azea-abs</i>)	74
	Corrigidos (<i>corrigidos</i>)	78
“Ruim”	Atípicos (<i>atípicos</i>)	18
	Estudantes (<i>estudantes</i>)	78

Estes córpus são descritos na Seção 3.2, e são utilizados com propósitos diferentes neste capítulo. Embora pareça intuitiva, não temos nenhum dado da avaliação completa da qualidade de escrita dos resumos para justificar a designação dos córpus em “bons” ou “ruins”. O que podemos esperar com relação ao córpus *azea-abs* é a ausência ou ínfima quantidade de erros gramaticais, pois retrata textos publicados e possuem boa organização esquemática, pois foram avaliados quanto à qualidade desta estrutura (presença de componentes essenciais, organização e balanceamento entre componentes).

Quanto ao córpus *estudantes*, nada podemos saber quanto à qualidade da estrutura esquemática, mas todos seus textos possuem problemas gramaticais. Desta forma, os textos do córpus *corrigido* são garantidos estarem sem erros gramaticais. O córpus *atípicos* possui problemas quanto à qualidade da estrutura esquemática, mas por serem publicados espera-se não possuírem erros gramaticais.

Foram treinados doze classificadores em três experimentos, usando cada um quatro diferentes algoritmos do WEKA: *Naive Bayes* (NB), J48, JRip e SMO:

- com os córpus *azea-abs* X *atípicos*, sendo que as informações sobre as estruturas esquemáticas de cada texto foram dadas com a anotação manual;
- com os mesmos córpus de (a), entretanto usando a classificação dada pelo AZEA para as sentenças do resumo (cenário real de uso do detector de qualidade);

(c) juntando os *córpus azea-abs* com *corrigidos* e *atípicos* com *estudante*.

Este último experimento se propôs a avaliar a influência do tamanho do *córpus* e a variação dos parâmetros de qualidade, isto é, parte do *córpus* com textos classificados como “bom” recebia este rótulo por razões diferentes: qualidade de estrutura esquemática e ausência de erros gramaticais. O mesmo se dá com o *córpus* com resumos classificados com “ruim”.

7.4.1 Atributos

O conjunto de atributos utilizados foi composto por medidas obtidas com a aplicação das dimensões já implementadas da rubrica (D1, D2 e parte da D5), medidas superficiais do Sistema de Larkey (LARKEY, 1998) e uma métrica de redes complexas.

Rubrica: Três das dimensões da rubrica foram modeladas como atributos. As dimensões D1 e D2 originaram, respectivamente, cinco e três atributos. Esses atributos são booleanos, sendo dada a presença ou ausência dos possíveis quesitos de avaliação de cada dimensão, descritos na Seção 7.1.

A dimensão D5 foi representada parcialmente pelo erro de uso de artigo. O valor deste atributo é a porcentagem de erros de uso de artigo, por SNs.

Larkey : Foram implementadas as onze *Text-complexity features* do sistema de Larkey, descritas na Seção 5.3.3. A implementação segue a descrição de cada um dos atributos, que podem ser vistos na Tabela 26 do Capítulo 5.

Redes complexas : Foi extraída a métrica “desvio da componente conexa” (DCC), de redes complexas, que possui boa correlação com a qualidade de textos (ANTIQUERA et al., 2005; ANTIQUERA, 2007; ANTIQUERA et al., 2007).

Na Tabela 48, descrevemos os 21 atributos utilizados pelos 12 classificadores treinados

7.4.2 Extração dos atributos

Pré-processamento

O pré-processamento é a etapa mais custosa, é feito por diversas ferramentas de PLN. O processo completo de pré-processamento e extração de atributos é esquematizado

Tabela 48: Atributos utilizados pelos classificadores binários da qualidade de resumos

Tipo	Atributo	Descrição	Valores
D1	D1A	Problemas na organização de sentenças	Sim (Y) ou Não (N)
	D1B	Falta componente: <i>purpose</i>	
	D1C	Falta componente: <i>result</i>	
	D1D	Falta componente: <i>conclusion</i>	
	D1E	Componente ' <i>gap</i> ' existe, mas não está seguida pelo propósito	
D2	D2A	Componente ' <i>purpose</i> ' deve existir, escrita em uma única sentença	Sim (Y) ou Não (N)
	D1B	Componente ' <i>conclusion</i> ' deve existir, escrita em uma única sentença	
	D2C	Componente ' <i>background</i> ' excede, em número de palavras, 30% do resumo	
D5	ArtAcc	Porcentagem de acerto do uso de artigos	0,0 ... 100
Larkey	L1	Número de caracteres no documento	Numérico
	L2	Número de palavra no documento	
	L3	Número de palavras distintas no documento	
	L4	A raiz quarta do número de palavras no documento	
	L5	Número de sentenças no documento	
	L6	Tamanho médio das palavras ($L1/L2$)	
	L7	Tamanho médio das sentenças, em número de palavras ($L2/L5$)	
	L8	Número de palavras com mais de 5 caracteres	
	L9	Número de palavras com mais de 6 caracteres	
	L10	Número de palavras com mais de 7 caracteres	
	L11	Número de palavras com mais de 8 caracteres	
Redes Complexas	DCC	Desvio da componente conexa	Numérico

na Figura 33. Os textos são processados pelas seguintes ferramentas de PLN: sentenciador (*OpenNLP*), tokenizador (*Penn Treebank*), etiquetador morfossintático (MXPOST) e *chunker* (*OpenNLP*). Os primeiros atributos extraídos são os 11 do sistema de Larkey, logo após a tokenização. Em seguida, o AZEA é aplicado e extrai a estrutura esquemática do resumo. O classificador usado foi o induzido pelo algoritmo SMO sobre o cópulo de treinamento. Neste ponto, são aplicadas as dimensões D1 e D2 da rubrica à estrutura esquemática do resumo. Os erros detectados, que têm atributos homônimos, recebem “Y”.

A modelagem em redes complexas e a extração da métrica DCC é feita por ferramentas desenvolvidas no NILC durante o mestrado de Antiqueira (2007), e apenas precisa do texto lematizado e sem *stopwords*. Para o atributo *ArtAcc*, primeiramente os SNs do texto são processados para se extrair os valores dos atributos do detector de erros de uso

Figura 33: Pré-processamento e extração dos valores dos atributos

de artigo, descritos na Seção 7.3.2. Depois, os SNs são rotulados pelo classificador ternário de erros de uso de artigo treinado com o algoritmo J48 (descrito na Seção 7.3.3), com o WEKA. O valor do atributo *ArtAcc* é a taxa de acerto apurada na classificação dos SNs do texto.

7.4.3 Treinamento e avaliação

O ambiente WEKA foi usado na indução e avaliação dos doze classificadores, sendo usados quatro algoritmos para os testes: J48, JRip, *Naive Bayes* (NB) e SMO. Na avaliação, o método para estimar o erro real dos classificadores foi o *10 fold stratified cross-validation*. As tabelas 49, 51 e 52 reportam, para cada um dos indutores utilizados, os valores de taxa de acerto e *Kappa*, além das métricas precisão (**P**), cobertura (**C**) e medida-F (**F**) para cada as classes (*bom* e *ruim*).

Iniciamos os experimentos usando os corpú *azea-abs* e *atípicos* para o treinamento. Usamos a estrutura esquemática anotada manualmente em vez de usarmos o AZEA para tal

tarefa. Os resultados são mostrados na Tabela 49 e foram estimados com o método *10 fold stratified cross-validation* no treinamento.

Tabela 49: Valores das métricas para com a estrutura esquemática real utilizada na extração dos atributos.

Algoritmo	Acerto	<i>Kappa</i>	Classe <i>bom</i>			Classe <i>ruim</i>		
			P	C	F	P	C	F
J48	78,26%	0,21	84%	90%	87%	42%	28%	33%
JRip	83,70%	0,42	87%	93%	90%	62%	44%	52%
NB	79,35%	0,38	89%	85%	87%	48%	56%	51%
SMO	83,70%	0,29	84%	99%	91%	80%	22%	35%
<i>baseline</i>	80,43%	0,00	80%	100%	89%	0%	0%	0%

As taxas de acerto estão bem próximas da *baseline*, embora alguns valores de *Kappa* estejam acima. Um fator que dificulta a indução dos classificadores é a pouca quantidade de resumos, apenas 94. A árvore de decisão gerada pelo classificador induzido com o algoritmo J48 é mostrada na Figura 34. Esta árvore consegue separar todos os exemplos dos casos de treinamento, isto é, testando sobre o mesmo corpus de treino, essa árvore prediz corretamente todos os casos. Portanto, a árvore de decisão da Figura 34 mostra, segundo nossos atributos, os critérios para a designação de um resumo do corpus *azea-abs* ou *atípicos*.

```

D1C = N
|   L5 <= 5
|   |   D2C = N
|   |   |   ArtAcc <= 76.19: bad (3.0)
|   |   |   ArtAcc > 76.19: good (5.0)
|   |   D2C = Y: bad (3.0)
|   L5 > 5
|   |   ArtAcc <= 80.49
|   |   |   ArtAcc <= 80
|   |   |   |   L10 <= 65: good (16.0)
|   |   |   |   L10 > 65
|   |   |   |   |   ArtAcc <= 67.86: bad (3.0)
|   |   |   |   |   ArtAcc > 67.86
|   |   |   |   |   |   L7 <= 21.25: bad (2.0)
|   |   |   |   |   |   L7 > 21.25: good (8.0)
|   |   |   |   |   ArtAcc > 80: bad (3.0)
|   |   |   |   ArtAcc > 80.49: good (45.0)
D1C = Y: bad (4.0)

```

Figura 34: Árvore de decisão gerada pelo classificador J48, com a estrutura esquemática anotado manualmente.

Ainda na Figura 34, podemos observar os atributos utilizados na classificação. Temos

atributos de quase todos os grupos, com exceção do único atributo de redes complexas, que não foi utilizado na árvore. Buscamos então medir a contribuição de cada um dos grupos de atributos na classificação. Para isso, fizemos testes alterando o conjunto de atributos usado na indução de duas maneiras: removendo-os e usando-os sozinhos.

Na Tabela 50 são mostrados os valores de *Kappa* como uma estimativa da contribuição individual dos grupos de atributos, para os quatro classificadores induzidos por diferentes algoritmos. Na segunda coluna temos o valor do *Kappa* quando o grupo de atributos é desconsiderado, e na terceira coluna o *Kappa* para o classificador somente com o atributo indicado no início linha. Na última linha, temos os valores para o grupo *Rubrica*, que é composto pelos atributos dos grupos D1, D2 e o atributo *ArtAcc*. Esse grupo treinado com o NB obteve *Kappa* de 0,47, superior ao próprio *Kappa* desse classificador com todos os atributos.

Tabela 50: Contribuição dos atributos. Valores do *Kappa*.

Grupo de Atributos	Atributos excluídos				Atributos sozinho			
	J48	JRip	NB	SMO	J48	JRip	NB	SMO
<i>D1</i>	0,00	0,00	0,13	0,06	0,31	0,31	0,29	0,31
<i>D2</i>	0,31	0,26	0,14	0,17	0,02	0,29	0,14	0,00
<i>ArtAcc</i>	0,31	0,29	0,20	0,22	0,00	0,00	0,00	-0,06
<i>Larkey</i>	0,31	0,29	0,38	0,17	0,00	0,00	0,17	0,06
<i>Redes Complexas</i>	0,31	0,35	0,18	0,22	0,00	0,00	0,00	0,00
<i>Rubrica</i>	0,00	0,00	0,13	0,06	0,31	0,31	0,47	0,19

O *Kappa* obtido pelo classificador treinado somente com os grupos de atributos (descritos na Tabela 48) indica que os atributos da dimensão D1, seguido pelos atributos de D2, são os grupos de atributos mais preditivos. Acreditamos que isso aconteça devido ao fato de que o critério mais importante que distingue o *córpus azea* do *córpus atípicos* seja a organização da estrutura. Outro fator pode ser a qualidade de anotação da estrutura esquemática, pois os atributos das dimensões D1 e D2 são os obtidos da estrutura esquemática anotada manualmente.

No segundo experimento, testamos então a influência da automatização da detecção da estrutura esquemática na classificação. Refizemos os testes, agora com a estrutura esquemática obtida pelo AZEA. É importante ressaltar que, uma vez que o AZEA foi treinado com o *córpus azea*, o acerto na extração da estrutura esquemática é superior ao informado⁸. A Tabela 51 mostra os valores obtidos nesse experimento.

⁸O AZEA (SMO) obteve 80,5% de acerto e *Kappa* de 0,73, estimado pelo *10 fold cross-validation*. Já seu acerto aparente, que é a taxa de acerto do classificador testado sobre o mesmo *córpus* utilizado no treino, foi de 87,4% e *Kappa* de 0,82.

Tabela 51: Valores das métricas para o experimento com estrutura esquemática obtida pelo AZEA

Algoritmo	Acerto	<i>Kappa</i>	Classe <i>bom</i>			Classe <i>ruim</i>		
			P	C	F	P	C	F
J48	75,00%	0,22	85%	84%	84%	37%	39%	38%
JRip	82,61%	0,47	90%	88%	89%	55%	61%	58%
NB	78,26%	0,39	90%	82%	86%	46%	61%	52%
SMO	82,61%	0,17	82%	100%	90%	100%	11%	20%
<i>baseline</i>	80,43%	0,00	80%	100%	89%	0%	0%	0%

Os valores obtidos com a estrutura esquemática informada pelo AZEA (Tabela 51) são semelhantes aos obtidos pela estrutura esquemática anotada manualmente. Entretanto, os resultados das taxas de acerto dos classificadores estão muito próximos aos valores obtidos pela *baseline*, independente da estrutura esquemática ter sido obtida de forma manual ou automática.

Aplicamos os classificadores cuja estrutura esquemática manual fora utilizada na extração dos atributos sobre os resumos dos corpúscos *corrigidos* e *estudantes*. Os resultados são muito baixos, muito próximos da *baseline*, como pode ser observado na Tabela 52.

Algoritmo	Acerto	<i>Kappa</i>	Classe <i>bom</i>			Classe <i>ruim</i>		
			P	C	F	P	C	F
J48	51,28%	0,03	60%	8%	14%	51%	95%	66%
JRip	50,64%	0,01	50%	64%	56%	51%	37%	43%
NB	50,64%	0,01	52%	15%	24%	50%	86%	64%
SMO	49,36%	-0,01	47%	9%	15%	50%	90%	64%
<i>baseline</i>	50,00%	0,00	50%	100%	67%	0%	0%	0%

Tabela 52: Resultados dos classificadores treinados com os corpúscos *azea* e *atípicos* e aplicados ao corpúscos de *corrigidos* (bons) e *estudantes* (ruins)

Um dos possíveis motivos dos baixos valores obtidos é a composição dos corpúscos de alunos, uma vez que a estrutura do resumo corrigido é muito similar à estrutura do resumo escrito pelo estudante. Sendo assim, os atributos da rubrica das dimensões D1 e D2 são anulados, e também alguns atributos do grupo Larkey, como número de sentenças.

No terceiro e último experimento, todos os corpúscos são usados no treinamento, e a estrutura esquemática usada é a informada pelo AZEA; apenas o classificador NB obteve resultados de *Kappa* na faixa de 0,20. As métricas dos outros classificadores assemelham-se muito à *baseline*. Os resultados podem ser observados na Tabela 53.

Tabela 53: Valores das métricas para o treinamento com todos os corpos, e os atributos obtidos da estrutura esquemática obtida pelo AZEA.

Algoritmo	Acerto	$Kappa$	Classe <i>bom</i>			Classe <i>ruim</i>		
			P	C	F	P	C	F
J48	46,77%	-0,15	56%	60%	58%	29%	25%	27%
JRip	53,63%	-0,01	61%	68%	64%	38%	30%	34%
NB	61,69%	0,20	70%	66%	68%	50%	54%	52%
SMO	59,68%	0,04	62%	86%	72%	45%	18%	25%
<i>baseline</i>	61,29%	0,00	61%	100%	76%	0%	0%	0%

7.5 Considerações finais

Neste capítulo, descrevemos a implementação das duas dimensões D1 e D2 da rubrica para resumos científicos. Também descrevemos os estudos para a implementação da dimensão D5, especificamente a detecção de erros de uso de artigo em inglês. Primeiramente, descrevemos os trabalhos da literatura, sendo que nos baseamos principalmente no sistema desenvolvido por Han et al. (2006) para a definição do conjunto de atributos utilizados. Para o treinamento, foi construído um corpus de 723 resumos científicos das áreas de Física e Ciências Farmacêuticas. Mantivemos nossa abordagem de aplicação de vários algoritmos de aprendizado de máquina sobre o corpus, e nos aproveitamos do fato de que estudantes brasileiros dificilmente cometem erros de troca de artigos e propusemos um classificador binário, que determina se um SN deve conter ou não um artigo. Este classificador binário obteve 83,7% de precisão e $Kappa$ de 0,63.

Contudo, os resultados dos experimentos para a detecção de resumos de qualidade (“bons” resumos) foram ruins, embora para textos separados pela estrutura esquemática obtivemos resultados regulares. Acreditamos que estes resultados se devem ao fato de não possuirmos um corpus de resumos que: (i) tivesse sido avaliado quanto a sua qualidade de escrita, principalmente pelas dimensões da rubrica e (ii) fosse suficientemente grande e balanceado. Sem um corpus como este é difícil avaliar realmente as dimensões da rubrica.

8 *Conclusões*

Este trabalho teve como objetivo estudar, avaliar e implementar métodos de detecção automática da estrutura esquemática e métodos de avaliação da qualidade de escrita, ambos para resumo científicos em inglês. Nos propusemos a investigar a implementação de ferramentas deste dois tipos de métodos para serem usados em dois cenários: no *feedback* de ferramentas de suporte à escrita, como o SciPo-Farmácia, e no auxílio da construção de corpus de resumos de boa qualidade, também para as mesmas ferramentas de suporte à escrita.

Com o objetivo de implementar um detector automático da estrutura esquemática de resumos científicos em inglês, este trabalho primeiramente analisou outros trabalhos de detecção automática, tanto em inglês quanto em português. Usamos estes trabalhos para definir os atributos a serem utilizados. Também estudamos ferramentas de suporte à escrita para definirmos quais as características desejáveis em nosso detector.

Decidimos por utilizar uma abordagem exploratória, baseada na indução de diversos classificadores com diferentes tipos de algoritmos de aprendizado de máquina. Utilizamos os algoritmos mais tradicionais disponíveis no WEKA e com os melhores desempenhos nos problemas que tratamos. Também exploramos sistemas e técnicas usadas na avaliação automática da qualidade de escrita de dissertações, embora praticamente todos os sistemas fossem comerciais e difíceis de acoplar sem ter que reimplementar o método. O estudo desses sistemas também nos mostrou que os bons sistemas de avaliação automática de escrita são caros de produzir. Além disso, focamos nos sistemas que avaliam estilo, uma vez que consideramos a avaliação do conteúdo não adequada em nosso contexto de resumos científicos.

Um dos resultados desse trabalho foi a implementação de um detector automático da estrutura esquemática de resumos em inglês, denominado *AZEA*. Este detector foi baseado no *AZ* (TEUFEL, 1999; TEUFEL; MOENS, 2002), reimplementando alguns de seus atributos e incorporando outros novos. O *AZEA* foi treinado para classificar sentenças em uma das seis categorias possíveis dos resumos do SciPo-Farmácia: Contexto, Lacuna, Propósito,

Metodologia, Resultados e Conclusão. O *AZEA* obteve 80,4% de acerto e *Kappa* de 0,73, superior aos trabalhos correlatos descritos no Capítulo 4, embora seja difícil a comparação com estes sistemas por diferenças de língua, esquema de anotação e cópula. O valor de *Kappa* que obtivemos foi muito próximo ao relatado por Feltrim (2004) entre dois anotadores humanos, que foi de 0,69 para um esquema de anotação muito semelhante.

Entretanto, o *AZEA* quando usado na classificação de textos cuja a estrutura esquemática é atípica obteve forte queda, com valores de *Kappa* abaixo de 0,55. Isto indica uma forte dependência do *AZEA* aos atributos contextuais, que são os atributos com maior poder de predição, enquanto que no *AZPort* (FELTRIM, 2004) os melhores atributos são os de expressões-padrão. Apesar de trabalharmos especificamente na formulação de expressões-padrão, não conseguimos que estas fossem bastante preditivas. Um bom atributo de expressões-padrão poderia reduzir a queda de desempenho em textos com estrutura atípica, além de assimilarem a maneira humana de classificar as sentenças. Além disso, a formulação manual de expressões-padrão é cara, e consumiu o trabalho de uma lingüista experiente por quase dois meses. Um ponto a ser abordado em um trabalho futuro é o desenvolvimento de métodos computacionais para se extrair automaticamente expressões-padrão, nos moldes do trabalho de (ABDALLA; TEUFEL, 2006).

Testamos também um subconjunto, composto por oito atributos, para a classificação de outras seções de textos científicos, denominado *AZSections*. O melhor resultado foi para introduções (*Kappa*=0,66), cuja estrutura esquemática é bem similar à de resumos, seguido de conclusões (*Kappa*=0,48). Para as outras seções, os valores do *Kappa* foram abaixo de 0,40. Um dos fatores são os cortes (os cortes estão ajustados para resumos) utilizados na discretização dos atributos, uma vez que os textos são muito diferentes em tamanho dos resumos. Como trabalho futuro, sugerimos o ajuste dos cortes no próprio treinamento.

Sobre a avaliação da qualidade de escrita, focamos na implementação de dimensões da rubrica descrita no Capítulo 5. Duas dimensões (D1 e D2) foram implementadas, sendo baseadas fortemente no *AZEA*, pois essas dimensões são regras simples aplicadas sobre a estrutura esquemática. O detector de erros de uso de artigo, que faz parte da dimensão D5, obteve 83,7% de acerto e *Kappa* de 0,63 na tarefa de determinar se um artigo é ou não necessário, e 77,4% de acerto e 0,53 de *Kappa* para a aplicação dos dois classificadores binários. Pretendemos incorporar este classificador à interface *AZEA-Web*, embora seus resultados com os textos dos alunos não sejam satisfatórios.

O experimento para a categorização de resumos bons e ruins teve resultados não muito animadores, a não ser para os casos onde só a estrutura estava sendo avaliada. Para

a continuação destes estudos e, principalmente, para a implementação das outras dimensões da rubrica é necessário primeiro construir um *córpus*, com resumos anotados de acordo com a rubrica e uma nota ou classe, que pode ser simplesmente *bom* ou *ruim*. Além disso, poucos métodos da literatura foram explorados, assim como foi reduzido o número de métricas de redes complexas utilizadas. Alguns dos *feedbacks* fornecidos pela rubrica podem ser melhorados, especificamente o erro de “Problemas de organização”, que pode ser melhorado fornecendo alguma sugestão de alteração na estrutura esquemática. Ainda neste caso, poderíamos usar técnicas computacionais, como distância de edição, para indicar como eliminar o problema de organização das sentenças, com o mínimo de alterações necessárias.

Acreditamos que o ferramental intermediário, desenvolvido para a extração de atributos, seja também uma contribuição deste trabalho. Foram desenvolvidas várias ferramentas de modelagem para armazenar e manipular estruturas hierárquicas, e organizar e facilitar o acesso à *tokens*, etiquetas morfossintáticas, *chunks*, e sentenças. Também foram desenvolvidas ferramentas para a busca de expressões-padrão nos moldes descritos no Capítulo 6, detectores do núcleo do sintagma nominal e ferramentas que extraem os atributos sintáticos de verbos, *Tempo*, *Modal* e *Voz*. Entretanto, estas ferramentas precisam ser validadas, uma vez que apenas poucos testes foram feitos em nosso *córpus*. As ferramentas são bem fáceis de portar, uma vez que são modulares e estão escritas em Java ou ANSI C. Todas as ferramentas serão disponibilizadas no site do NILC.

Referências

- ABDALLA, R. M.; TEUFEL, S. A bootstrapping approach to unsupervised detection of cue phrase variants. In: *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*. Morristown, NJ, USA: Association for Computational Linguistics, 2006. p. 921–928.
- AIRES, R. V. X. *Implementação, Adaptação, Combinação e Avaliação de Etiquetadores para o Português do Brasil*. Dissertação (Mestrado) — ICMC-USP, 2000.
- ALBERT, R.; BARABASI, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, v. 74, p. 47, 2002. Disponível em: <<http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0106096>>.
- ALUÍSIO, S.; O.N, O. J. A case-based approach for developing writing tools aimed at non-native english users. *Lectures Notes in Artificial Intelligence*, v. 1010, p. 121–132, 1995.
- ALUÍSIO, S. M.; GANTENBEIN, R. E. Towards the application of systemic functional linguistics in writing tools. In: *Proceedings of International Conference on Computers and their Applications*. [S.l.: s.n.], 1997.
- ALUÍSIO, S. M.; JR., O. N. de O. A detailed schematic structure of research papers introductions: an application in suport-writing tools. *Revista de la Sociedad Espanyola para el Procesamiento del Lenguaje Natural*, v. 19, p. 141–147, 1996.
- ALUÍSIO, S. M.; SCHUSTER, E.; FELTRIM, V. D.; JR., A. P.; JR., O. N. O. Evaluating scientific abstracts with a genre-specific rubric. In: *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*. Amsterdam: [s.n.], 2005. v.1, p. 738–740.
- ANTHONY, L.; LASHKIA, G. Mover: A machine learning tool to assist in the reading and writing of technical papers. *IEEE Transactions on Professional Communication*, v. 46(3), p. 185–193, 2003.
- ANTIQUEIRA, L. *Desenvolvimento de Técnicas Baseadas em Redes Complexas para Sumarização Extrativa de Textos*. Dissertação (Mestrado) — ICMC-USP, São Carlos, SP, 2007.
- ANTIQUEIRA, L.; NUNES, M. das G. V.; JR., O. N. de O.; COSTA, L. da F. Modelando textos como redes complexas. In: *III TIL*. São Leopoldo, RS: [s.n.], 2005.
- ANTIQUEIRA, L.; NUNES, M. das G. V.; OLIVEIRA, O. N.; COSTA, L. da F. Strong correlations between text quality and complex networks features. *Physica A*, v. 373, p. 811–820, 2007.

BAEZA-YATES, R. A.; RIBEIRO-NETO, B. A. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999. ISBN 0-201-39829-X. Disponível em: <citeseer.ist.psu.edu/baeza-yates99modern.html>.

BARANAUSKAS, J.; MONARD, M. C. *Reviewing some machine learning concepts and methods*. [S.l.], 2000.

BARRASS, R. *Os Cientistas Precisam Escrever: Guia de Redação para Cientistas, Engenheiros e Estudantes*. São Paulo, SP: Editora da Universidade de São Paulo, 1979.

BURSTEIN, J.; CHODOROW, M.; LEACOCK, C. Criterionsm: Online essay evaluation: An application for automated evaluation of student essays. In: *Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*. [S.l.: s.n.], 2003. Acapulco, México.

BURSTEIN, J.; KUKICH, K.; WOLFF, S.; LU, C.; CHODOROW, M.; BRADEN-HARDER, L.; D., H. M. Automated scoring using a hybrid feature identification technique. In: *Annunal Meeting of the Association of Computational Linguistics*. [S.l.: s.n.], 1998. Montreal, Canada.

BURSTEIN, J.; LEACOCK, C.; SWARTZ, R. Automated evaluation of essays and short answers. In: *Proceedings of the 5th International Computer Assisted Assessment Conference (CAA 01)*. [s.n.], 2001. Disponível em: <citeseer.ist.psu.edu/699842.html>.

CALLAN, J. P.; CROFT, W. B.; BROGLIO, J. TREC and tipster experiments with inquiry. *Information Processing and Management*, v. 31, n. 3, p. 327–343, 1995. Disponível em: <citeseer.ist.psu.edu/callan94trec.html>.

CALLEAR, D.; JERRAMS-SMITH, J.; SOH, V. CAA of Short Non-MCQ Answers. In: DANSON, M.; C., E. (Ed.). *Proceedings of the 5th International CAA Conference*. Loughborough, UK: Loughborough University, 2001. Disponível em: <citeseer.ist.psu.edu/callear01caa.html>.

CHODOROW, M.; BURSTEIN, J. *Beyond essay length: Evaluating e-raters' performance on TOEFL essays*. Princeton, NJ., 2004.

CHRISTIE, J. R. Automated essay marking - for both style and content. In: DANSON, M.; SHERRAT, R. (Ed.). *Proceedings of the 3rd Annual CAA Conference*. Loughborough, UK: Loughborough University, 1999. p. 39–48. Disponível em: <citeseer.ist.psu.edu/christie99automated.html>.

COHEN, W. W. Fast effective rule induction. In: PRIEDITIS, A.; RUSSELL, S. (Ed.). *Proc. of the 12th International Conference on Machine Learning*. Tahoe City, CA: Morgan Kaufmann, 1995. p. 115–123. ISBN 1-55860-377-8. Disponível em: <citeseer.ist.psu.edu/cohen95fast.html>.

COLLINS, M. J. *Head-driven statistical models for natural language parsing*. Tese (Doutorado) — University of Pennsylvania, 1999. Supervisor-Mitchell P. Marcus.

CRAVEN, M.; DIPASQUO, D.; FREITAG, D.; MCCALLUM, A.; MITCHELL, T.; NIGAM, K.; SLATTERY, S. Learning to extract symbolic knowledge from the world wide web. In: *AAAI '98/IAAI '98: Proceedings of the fifteenth national/tenth conference on*

Artificial intelligence/Innovative applications of artificial intelligence. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998. p. 509–516. ISBN 0-262-51098-7.

DESSUS, P.; LEMAIRE, B.; VERNIER, A. *Free-text assessment in virtual campus*. 2000. Disponível em: <citeseer.ist.psu.edu/dessus00freetext.html>.

DUMAIS, S.; PLATT, J.; HECKERMAN, D.; SAHAMI, M. Inductive learning algorithms and representations for text categorization. In: *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*. New York, NY, USA: ACM Press, 1998. p. 148–155. ISBN 1-58113-061-9.

FELTRIM, V. D. *Uma abordagem baseada em córpus e em sistemas de crítica para a construção de ambientes Web de auxílio à escrita acadêmica em português*. Tese (Doutorado) — ICMC-USP São Carlos, SP, Outubro 2004.

FELTRIM, V. D.; ALUÍSIO, S. M.; NUNES, M. das G. V. Analysis of the rhetorical structure of computer science abstracts in portuguese. *Corpus Linguistics*, v. 16, p. 212–218, 2002. UCREL Technical Papers. Lancaster: Lancaster University.

FLEISS, J. L. *Statistical methods for rates and proportions*. 2nd edition. ed. [S.l.]: Jonh Wiley & Sons: New York, 1981.

FURNKRANZ, J.; WIDMER, G. Incremental reduced error pruning. In: *International Conference on Machine Learning*. [s.n.], 1994. p. 70–77. Disponível em: <citeseer.ist.psu.edu/article/furnkranz94incremental.html>.

GONÇALVES, L. S. M. *Categorização em Text Mining*. Dissertação (Mestrado) — ICMC-USP São Carlos, 2002.

GRAETZ, N. Teaching efl students to extract structural information from abstracts. In: *Ulijn and Pugh*. [S.l.]: (eds.), 1985.

HAN, N.-R.; CHODOROW, M.; LEACOCK, C. Detecting errors in english article usage by non-native speakers. *Nat. Lang. Eng.*, Cambridge University Press, New York, NY, USA, v. 12, n. 2, p. 115–129, 2006. ISSN 1351-3249.

HEARST, M. The debate on automated essay grading. *IEEE Intelligent Systems*, v. 15(5), p. 22–37, 2000.

JOACHIMS, T. Text categorization with support vector machines: learning with many relevant features. In: *Proceedings of the European Conference on Machine Learning (ECML)*. [S.l.]: Springer, 1998.

JR., L. G. G.; LIZZOTE, R.; SCHUSTER, E.; DAYRELL, C.; ALUÍSIO, S. M. A two-tiered approach to detecting english article usage: an application in scientific paper writing tools. Artigo submetido ao EMNLP 2007.

KEERTHI, S. S.; SHEVADE, S. K.; BHATTACHARYYA, C.; MURTHY, K. R. K. Improvements to platt's smo algorithm for svm classifier design. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 13, n. 3, p. 637–649, 2001. ISSN 0899-7667.

KNIGHT, K.; CHANDER, I. Automated postediting of documents. In: *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1994. p. 779–784. ISBN 0-262-61102-3.

KUKICH, K. Beyond automated essay scoring. *IEEE Intelligent Systems*, v. 15(5), p. 22–27, September/October 2000.

LANDAUER, T. K.; FOLTZ, P. W.; LAHAM, D. An introduction to latent semantic analysis. *Discourse Processes*, v. 25, p. 259–284, 1998.

LANDIS, J.; KOCH, G. The measurement of observer agreement for categorical data. *Biometrics*, v. 33, p. 159–174, 1977.

LARKEY, L. S. Automatic essay grading using text categorization techniques. In: *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.: s.n.], 1998. p. 90–95.

LEE, J. Automatic article restoration. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of ACL*. Boston, MA: [s.n.], 2004. p. 31–36.

LEWIS, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In: *ECML-98, 10th European Conference on Machine Learning*. (Chemnitz, Germany, 1998): [s.n.], 1998. p. 4–15.

MANNING, C. D.; SCHÜTZE, H. *Foundations of Statistical Natural Language Processing*. [S.l.]: Cambridge, Mass. : MIT Press, 1999. 712 p.

MARCUS, M. P.; MARCINKIEWICZ, M. A.; SANTORINI, B. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 19, n. 2, p. 313–330, 1993. ISSN 0891-2017.

MARÍN, D. P. *Automatic evaluation of users' short essays by using statistical and shallow natural language processing techniques*. Dissertação (Mestrado) — Universidad Autónoma de Madrid, 2004. 83p.

MARQUIAFÁVEL, V. *Um processo para a geração de recursos lingüísticos aplicáveis em ferramentas de auxílio à escrita científica*. Dissertação (Mestrado) — DL-UFSCar, São Carlos - SP, 2007. 280 p.

MASON, O.; GROVE-STEPHENSON, I. Automated free text marking with paperless school. In: *Proceedings of 6th Annual CAA Conference*. [S.l.: s.n.], 2002.

MING, P.; MIKHAILOV, A.; KUAN, T. L. Intelligent essay marking system. In: *Learners Together, Fev. 2000*. NgeeANN Polytechnic, Singapore: [s.n.], 2000.

MINNEN, G.; BOND, F.; COPESTAKE, A. Memory-based learning for article generation. In: *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*. Morristown, NJ, USA: Association for Computational Linguistics, 2000. p. 43–48.

- MITCHELL, T.; RUSSEL, T.; BROOMHEAD, P.; N., A. Towards robust computerised marking of free-text responses. In: *Proceedings of the Sixth International Computer Assisted Assessment Conference*. Loughboroug University, Loughborouh, UK: [s.n.], 2002.
- MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997.
- PAGE, E. B. The imminence of grading essays by computer. *Phi Delta Kappan*, p. 238–243, 1966.
- PAGE, E. B. New computer grading of student prose, using modern concepts and software. *Journal of Experimental Education*, v. 62(2), p. 127–142, 1994.
- PLATT, J. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*. [S.l.], 1998.
- QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1558602402.
- RATNAPARKHI, A. A maximum entropy part-of-speech tagger. In: *Empirical Methods in Natural Language Processing Conference*. [S.l.: s.n.], 1996.
- RATNAPARKHI, A. *A Simple Introduction to Maximum Entropy Models for Natural Language Processing*. [S.l.], 1997.
- ROSÉ, C. P.; ROQUE, A.; BHEMBE, D.; VANLEHN, K. A hybrid text classification approach for analysis of student essays. In: *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing*. Morristown, NJ, USA: Association for Computational Linguistics, 2003. p. 68–75.
- RUDNER, L. M.; LIANG, T. Automated essay scoring using bayes' theorem. *Journal of Technology, Learning, and Assessment*, v. 1, n. 2, p. 1–22, 2002.
- SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. *Commun. ACM*, ACM Press, New York, NY, USA, v. 18, n. 11, p. 613–620, 1975. ISSN 0001-0782.
- SANG, E. F. T. K.; BUCHHOLZ, S. Introduction to the conll-2000 shared task: chunking. In: *Proceedings of 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*. [S.l.: s.n.], 2000. p. 13–14, 2000, Lisbon, Portugal.
- SCHUSTER, E.; ALUÍSIO, S.; FELTRIM, V.; JR., A. P.; JR, O. N. O. Enhancing the writing of scientific abstracts: a two-phased process using software tools and human evaluation. In: *Encontro nacional de Inteligência Artificial (ENIA)*. [S.l.: s.n.], 2005. v.1, p. 962–971.
- SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys*, v. 34, n. 1, p. 1–47, 2002. Disponível em: <<http://faure.iei.pi.cnr.it/fabrizio/Publications/ACMCS02.pdf>>.
- SIEGEL, S.; CASTELLAN, N. *Nonparametric statistics for the behavioral sciences*. Second. [S.l.]: McGraw-Hill, Inc., 1988.

SUKKARIEH, J.; PULMAN, S.; RAIKES, N. Auto-marking: Using computational linguistics to score short, free text responses. In: *Proceedings of the 29th Annual Conference of the International Association for Educational Assessment*. Manchester, U.K.: [s.n.], 2003.

SWALES, J. *Genre Analysis: English in Academic and Research Settings*. [S.l.]: Cambridge Press, 1990.

TEUFEL, S. *Argumentative Zoning: Information Extraction from Scientific Text*. Tese (Doutorado) — School of Cognitive Science, University of Edinburg, Edinburg, UK, 1999.

TEUFEL, S.; CARLETTA, J.; MOENS, M. An annotation scheme for discourse-level argumentation in research articles. In: *Ninth Meeting of the European Chapter of the Association for Computational Linguistics (EACL-99)*. [S.l.: s.n.], 1999. p. 110–117.

TEUFEL, S.; MOENS, M. Summarising scientific articles - experiments with relevance and rhetorical status. *Computational Linguistics*, v. 28 (4), p. 409–446, 2002.

VALENTI, S.; NERI, F.; CUCCHIARELLI, A. An overview of current research on automated essay grading. *Journal of Information Technology Education*, v. 2, p. 1–12, 2003.

VAPNIK, V. N. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

WEISSBERG, R.; BUKER, S. *Writing up Research: Experimental Research Report Writing for Students of English*. [S.l.]: Prentice Hall, 1990.

WILLIAMS, R. Automated essay grading: an evaluation of four conceptual models. In: *Proceedings of the 10th Annual Teaching Learning Forum*. [S.l.: s.n.], 2001.

WITTEN, I. H.; FRANK, E. *Data Mining: Practical machine learning tools and techniques*. 2nd. ed. [S.l.]: Morgan Kaufmann, 2005.

APÊNDICE A – Telas do Criterion (BURSTEIN et al., 2001, 2003)

Figura 35: Resumo do desempenho de um aluno

critload - Mozilla Firefox

Arquivo Editar Exibir Ir Favoritos Ferramentas Ajuda

http://www.ets.org/Media/Products/criterion/tour1/index.html

Últimas notícias Inglês -> Português Dúvidas Firefox

ETS Criterion Student: Danielle Cameron Help Log Out

Click items in this row to go back Home Assignment: Lengthening the School Year

Performance Summary

Essay Assignment: Lengthening the School Year
Suggested Time: 25 minutes
Time Taken: 20 minutes

Holistic Score: **4** [View Score Analysis](#)

Look carefully at the **grammar, usage, and mechanics** feedback for this essay and check the "Writer's Handbook" for each feedback message. Working on the grammar, usage, and mechanics in the essay may help you improve it.

Trait Feedback Analysis

Grammar:	1 error	View Grammar results
Usage:	1 error	View Usage results
Mechanics:	5 errors	View Mechanics results
Style:	3 comments	View Style comments
Organization & Development:		View Organization & Development comments

Click the **Continue** button to return to your Home page.

Continue

Begin | Students | **Feedback** | Teachers | Assignments | Reports | End 68/607 seconds **RELOAD**

Figura 36: *Feedback* de organização

*APÊNDICE B – Erros gramaticais levantados
na formulação da dimensão
D5 da rubrica de resumos
científicos (JR. et al.,)*

Tipo de erro	Descrição do erro	Exemplos	Correção Sugerida
ART	substitution of one article for another	to get common sense from a general public and	to get common sense from the general public and
ART -	missing an article needed in English	quality of the sausage	the quality of the sausage
ART+	having an article not needed in English	The molecular markers are employed...	Molecular markers are employed...
CAP	capitalization problem	brazilian fishing families	Brazilian fishing families
P	punctuation problem	. Also	. Also,
PORT	using a Portuguese word in the English	aquoso	aqueous
POS	part of speech error	a sausage common consumed in different regions	a sausage commonly consumed in different regions
S/PL	singular/plural noun error	difference	differences
S/PL ADJ	using the plural for an adjective	others variables	other variables
SP	spelling error	bledding	bleeding
SVA	subject-verb agreement error	Human contact with blisters cause	Human contact with blisters causes
S-VO	missing subject	was detected ..S. aureas in sixteen days...	S. aureus was detected in sixteen days...
S+VO	extra subject	It was employed the methodology of experimental design.	The methodology of experimental design was employed.
SV-O	missing verb	In the following concentrations: 5.0...	It was effective in the following concentrations: 5.0...
VF	error in the form of the verb	are find	are found
VU	error in the use of the verb (tense)	are found	were found
WO	word order error	that only food ingested during the day was not enough	that food ingested only during the day was not enough
WO ADJ	word order error in adjectives	for loci polymorphic	for polymorphic loci
WO NP	word order error in complex NPs	experimental design methodology	the methodology of experimental design
WO S	incorrect word order for the subject in main clause	It was employed the methodology of experimental design .	The methodology of experimental design was employed.
WU	word use (lexical) error	pledge	guarantee
WF	word form error	this three genes	these three genes
WU Col	word use (lexical) error collocation	capable to produce	capable of producing

Figura 37: Tabela de categorização de erros da rubrica

	Tipo de erro	Número de erros	Porcentagem de erros
1.	WU	497	25,8%
2.	ART -	258	13,4%
3.	P	165	8,6%
4.	SP	147	7,6%
5.	WUCol	109	5,7%
6.	ART +	95	4,9%
7.	S/PL	74	3,8%
8.	SVA	69	3,6%
9.	VF	64	3,3%
10.	S/PL_ADJ	50	2,6%
11.	WF	48	2,5%
12.	WO_ADJ	48	2,5%
13.	POS	47	2,4%
14.	VU	47	2,4%
15.	WO_NP	38	2,0%
16.	CAP	34	1,8%
17.	Port	34	1,8%
18.	WO_S	33	1,7%
19.	S+VO	19	1,0%
20.	WO	18	0,9%
21.	ART	17	0,9%
22.	S-VO	9	0,5%
23.	SV-O	5	0,3%
	TOTAL	1925	100%

Figura 38: Distribuição dos erros da dimensão D5 nos resumos dos alunos. Os 6 erros mais comuns estão destacados

*ANEXO A – Etiquetas morfossintáticas
utilizadas no corpus Penn
Treebank (MARCUS et al., 1993)*

Table 2
The Penn Treebank POS tagset.

1. CC	Coordinating conjunction	25. TO	<i>to</i>
2. CD	Cardinal number	26. UH	Interjection
3. DT	Determiner	27. VB	Verb, base form
4. EX	Existential <i>there</i>	28. VBD	Verb, past tense
5. FW	Foreign word	29. VBG	Verb, gerund/present participle
6. IN	Preposition/subordinating conjunction	30. VBN	Verb, past participle
7. JJ	Adjective	31. VBP	Verb, non-3rd ps. sing. present
8. JJR	Adjective, comparative	32. VBZ	Verb, 3rd ps. sing. present
9. JJS	Adjective, superlative	33. WDT	<i>wh</i> -determiner
10. LS	List item marker	34. WP	<i>wh</i> -pronoun
11. MD	Modal	35. WP\$	Possessive <i>wh</i> -pronoun
12. NN	Noun, singular or mass	36. WRB	<i>wh</i> -adverb
13. NNS	Noun, plural	37. #	Pound sign
14. NNP	Proper noun, singular	38. \$	Dollar sign
15. NNPS	Proper noun, plural	39. .	Sentence-final punctuation
16. PDT	Predeterminer	40. ,	Comma
17. POS	Possessive ending	41. :	Colon, semi-colon
18. PRP	Personal pronoun	42. (	Left bracket character
19. PP\$	Possessive pronoun	43.)	Right bracket character
20. RB	Adverb	44. "	Straight double quote
21. RBR	Adverb, comparative	45. '	Left open single quote
22. RBS	Adverb, superlative	46. "	Left open double quote
23. RP	Particle	47. '	Right close single quote
24. SYM	Symbol (mathematical or scientific)	48. "	Right close double quote

ANEXO B – Rubrica usada na avaliação do GMAT (AWA Scoring Guide)

A score of 6 is Outstanding– A cogent, well-articulated analysis of the complexities of the issue, demonstrating mastery of the elements of effective writing, and displaying the following characteristics:

- explores ideas and develops a position on the issue with insightful reasons and/or persuasive examples is clearly well organized
- demonstrates superior control of language, including diction and syntactic variety
- demonstrates superior facility with the conventions of standard written English, but may have minor flaws

A score of 5 is Strong– A well-developed analysis of the complexities of the issue, demonstrating strong control of the elements of effective writing, and displaying the following characteristics:

- explores ideas and develops a position on the issue with well-chosen reasons and/or examples
- is generally well organized
- demonstrates clear control of language, including diction and syntactic variety
- demonstrates facility with the conventions of standard written English, but may have minor flaws

A score of 4 is Adequate– A competent analysis of the issue, demonstrating adequate control of the elements of effective writing, and displaying the following characteristics:

- develops a position on the issue with relevant reasons and/or examples
- is adequately organized

- demonstrates adequate control of language, including diction and syntactic variety, but may lack syntactic variety
- displays control of the conventions of standard written English, but may have some flaws

A score of 3 is Limited– A competent analysis of the issue, demonstrating adequate control of the elements of effective writing, and displaying the following characteristics:

- is vague or limited in developing a position on the issue
- is poorly organized
- is weak in the use of relevant reasons or examples
- uses language imprecisely and/or lacks sentence variety
- contains occasional major errors or frequent minor errors in grammar, usage, and mechanics

A score of 2 is Seriously Flawed– A paper demonstrating serious weakness in analytical writing skills, and displaying the following characteristics:

- is unclear or seriously limited in presenting or developing a position on the issue
- is disorganized
- provides few, if any, relevant reasons or examples
- has serious, frequent problems in the use of language and sentence structure
- contains numerous errors in grammar, usage, or mechanics that interfere with meaning

A score of 1 is Fundamentally Deficient– A paper demonstrating fundamental deficiencies in analytical writing skills, and displaying the following characteristics:

- provides little evidence of the ability to develop or organize a coherent response to the topic
- has severe and persistent errors in language and sentence structure
- contains a pervasive pattern of errors in grammar, usage, and mechanics that severely interferes with meaning

A score of 0 is Unscorable– A paper that is totally illegible or obviously not written on the assigned topic.