Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Métodos para Análise Discursiva Automática

Thiago Alexandre Salgueiro Pardo

Orientadora: Profa. Dra. Maria das Graças Volpe Nunes

SÃO CARLOS

JUNHO, 2005

“Somente o melhor pode acontecer.”

Henry McCoy

AGRADECIMENTOS

A minha família, pelo apoio e amor incondicionais.

Aos colegas do NILC e agregados, pelo companheirismo, pelos bate-papos, pelas discussões científicas, pela preciosa amizade.

Aos amigos que, longe ou perto, estão sempre presentes: Ariani, Anselmo, Andréia, Leandro, Helena, Raquel, Lola e Fábio.

A Daniel Marcu, pela supervisão, boas idéias e "ganância científica" que tanto contribuíram para este trabalho.

A minha querida orientadora, Graça, uma graça de pessoa, pela orientação pessoal e profissional, pelas críticas valorosas e sempre construtivas, pelo profissionalismo, pelo suporte e apoio de todas as horas, pelo otimismo, pela amizade. A quem admiro muito, pela força, pelo empenho desmedido pelo NILC e pelo PLN no Brasil, por lidar com tudo e com todos de forma nunca menos que formidável, por sempre tocar o barco para frente e, além de tudo isso e mesmo com tudo isso, por nunca faltar a uma *happy hour*. Uma referência que levarei por toda a vida.

Às agências de fomento à pesquisa FAPESP, CAPES, CNPq e Comissão Fulbright.

ÍNDICE GERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

RESUMO

Pesquisas em Lingüística e Lingüística Computacional têm comprovado há tempos que um texto é mais do que uma simples seqüência de sentenças justapostas. Um texto possui uma estrutura subjacente altamente elaborada que relaciona todo o seu conteúdo, atribuindo-lhe coerência. A essa estrutura dá-se o nome de estrutura discursiva, sendo ela objeto de estudo da área de pesquisa conhecida como Análise de Discurso. Diante da grande utilidade desse conhecimento para diversas aplicações de Processamento de Línguas Naturais, por exemplo, sumarização automática de textos e resolução de anáforas, a análise discursiva automática tem recebido muita atenção. Para o português do Brasil, em particular, há poucos recursos e pesquisas nessa área de pesquisa. Neste cenário, esta tese de doutorado visa a investigar, desenvolver e implementar métodos para análise discursiva automática, adotando como principal teoria discursiva a *Rhetorical Structure Theory*, uma das teorias mais difundidas atualmente. A partir da anotação retórica e da análise de um corpus de textos científicos da Computação, produziu-se o primeiro analisador retórico automático para a língua portuguesa do Brasil, chamado DiZer (*DIscourse analyZER*), além de uma grande quantidade de conhecimento discursivo. Apresentam-se modelos estatísticos inéditos para o reconhecimento de relações discursivas baseados em unidades de conteúdo de crescente complexidade, abordando palavras, conceitos e estruturas argumentais. Em relação a este último item, é apresentado um modelo para o aprendizado não supervisionado das estruturas argumentais dos verbos, o qual foi aplicado para os 1.500 verbos mais freqüentes do inglês, resultando em um repositório chamado ArgBank. O DiZer e os modelos propostos são avaliados, produzindo resultados satisfatórios.

ABSTRACT

Researches in Linguistics and Computational Linguistics have shown that a text is more than a simple sequence of juxtaposed sentences. Every text contains a highly elaborated underlying structure that relates its content, attributing coherence to the text. This structure is called discourse structure and is the object of study in the research area known as Discourse Analysis. Given the usefulness of this kind of knowledge for several Natural Language Processing tasks, e.g., automatic text summarization and anaphora resolution, automatic discourse analysis became a very important research topic. For Brazilian Portuguese, in particular, there are few resources and researches about it. In this scenario, this thesis aims at investigating, developing and implementing methods for automatic discourse analysis, following the Rhetorical Structure Theory mainly, one of the most used discourse theories nowadays. Based on the rhetorical annotation and analysis of a corpus of scientific texts from Computers domain, the first rhetorical analyzer for Brazilian Portuguese, called DiZer (DIscourse analyZER), was produced, together with a big amount of discourse knowledge. Novel statistical models for detecting discourse relations are presented, based on content units of increasing complexity, namely, words, concepts and argument structures. About the latter, a model for unsupervised learning of verb argument structures is presented, being applied to the 1.500 most frequent English verbs, resulting in a repository called ArgBank. DiZer and the proposed models are evaluated, producing satisfactory results.

1. Introdução

Ao ler um texto, nós, humanos, realizamos mais do que a interpretação de cada sentença isoladamente. Tentamos relacionar o conteúdo das sentenças e de suas partes para que o texto faça sentido como um todo e, portanto, seja coerente. De fato, pesquisas em Lingüística e Lingüística Computacional têm comprovado há tempos que um texto coerente possui uma estrutura altamente elaborada, no nível discursivo, subjacente à superfície textual.

Segundo Koch e Travaglia (2002), um texto é coerente se é possível lhe atribuir um sentido global em uma situação de comunicação. Para haver um sentido global, deve ser possível determinar relações entre os conteúdos expressos pelos segmentos textuais, ou seja, entre suas proposições. É essa propriedade que diferencia o trecho de texto (1) de (2) abaixo.

(1) "Embora tenha chovido, as obras continuaram."
(2) "João não foi à aula, mas estava doente."

Em cada um destes trechos, tem-se uma aparente relação de oposição entre as proposições expressas pelas orações, mas apenas o trecho (1) é coerente. Em (1), é possível determinar um sentido global. Em (2), por outro lado, há incoerência, pois a relação de oposição contraria a relação de causa que parece mais plausível (isto é, aceitável). Dessa forma, apesar das duas orações em (2) possuírem sentido individualmente, parece impossível estabelecer um sentido unitário para a seqüência.

Situação de comunicação refere-se ao próprio discurso. É no nível do discurso que um escritor, ao produzir um texto, organiza e relaciona as proposições veiculadas por este a fim de atingir determinados objetivos comunicativos, isto é, satisfazer suas intenções, como persuadir o leitor a realizar uma ação ou informar o leitor sobre algo, por exemplo. Considere o trecho de texto (3) abaixo:

(3) "Embora você não goste, trabalhar é importante. O trabalho enobrece o homem."

Ao interpretar este trecho, pode-se perceber que:

- o objetivo comunicativo do escritor é persuadir uma pessoa a trabalhar;

- as proposições expressas pelos trechos “Embora você não goste” e “trabalhar é importante” estão em uma relação de oposição;
- a proposição expressa pelo trecho “trabalhar é importante” é mais importante do que a expressa pelo trecho “Embora você não goste” para a satisfação do objetivo comunicativo pretendido pelo escritor do texto;
- a proposição expressa pelo trecho “O trabalho enobrece o homem” justifica a afirmação “trabalhar é importante”;
- a proposição expressa pelo trecho “trabalhar é importante” é mais importante do que a expressa pelo trecho “O trabalho enobrece o homem”.

Além disso, o objetivo de persuadir uma pessoa a trabalhar pode ser derivado dos objetivos subjacentes aos trechos “Embora você não goste” e “trabalhar é importante”, que poderiam ser, respectivamente, “reconhecer o fato de a pessoa não gostar de trabalhar” e “ressaltar a importância do trabalho”. É possível perceber também que o objetivo subjacente ao segundo trecho é dominante em relação ao do primeiro trecho para a satisfação do objetivo primário de “persuadir uma pessoa a trabalhar”.

Como se pode notar, há diversos aspectos envolvidos na interpretação de um texto em uma determinada situação discursiva, aspectos estes que, em geral, não são percebidos isoladamente por um humano durante a leitura. De forma sistemática, é possível identificar: objetivos comunicativos, ou seja, as intenções do escritor do texto ao produzi-lo, tanto para o texto como um todo quanto para suas partes; relações entre estas intenções, indicando quais intenções são mais importantes para a satisfação do objetivo primário de um texto; relações intencionais/argumentativas entre proposições, cujo efeito pretendido é aumentar a inclinação do leitor para alguma coisa, alterando sua crença em algo, fazendo-o desejar realizar uma ação ou aceitar melhor algo (por exemplo, justificar uma afirmação, para que o leitor aceite melhor esta); relações semânticas/informativas/factuais, cujo efeito pretendido é fazer com que o leitor reconheça/fique informado sobre a relação (por exemplo, o fato de uma pessoa estar doente implicar o fato de ela não ir à aula).

Em geral, os aspectos distintos do discurso são tratados por diferentes teorias discursivas. Por teorias discursivas, refere-se a teorias propostas que visam a explicar e/ou estruturar o discurso segundo princípios específicos. Por exemplo, Grosz e

Sidner (1986) visam a modelar o aspecto intencional do discurso na teoria que propõem; Jordan (1992) e Kehler (2002) sugerem conjuntos de relações semânticas para a estruturação textual; Mann e Thompson (1987) propõem uma teoria de estruturação retórica de textos, chamada *Rhetorical Structure Theory* (RST), que engloba relações intencionais e semânticas. Algumas pesquisas também propõem formas de unificar algumas das teorias discursivas existentes (por exemplo, Moore e Pollack, 1992; Korelsky e Kittredge, 1993; Hovy, 1993; Maier, 1993; Dale, 1993; Moore e Paris, 1993; Moser e Moore, 1996; Rino, 1996; Marcu, 1999, 2000a). A RST, em particular, é uma das teorias discursivas de maior impacto (Reiter e Dale, 2000), sendo utilizada para os mais diversos fins, de análise à geração textual automática.

O uso de conhecimento discursivo, em seus vários níveis, é de grande valia para sistemas de Processamento de Língua Natural (PLN). PLN, segundo Dias da Silva (1996), constitui um domínio complexo e multifacetado, cujo objetivo é a projeção e implementação de sistemas computacionais que processam língua natural, por exemplo, corretores ortográficos e gramaticais, ferramentas de auxílio à escrita, sumarizadores de texto, tradutores automáticos e sistemas de diálogo. Pesquisas têm demonstrado a contribuição que o conhecimento discursivo pode oferecer: sistemas de diálogo são capazes de identificar os objetivos dos participantes do diálogo e interagir de forma mais apropriada (veja, por exemplo, Moore e Paris, 1993; Moore, 1995); sumarizadores de texto são capazes de identificar os segmentos textuais mais importantes de forma mais informada para compor o sumário correspondente (veja, por exemplo, O'Donnel, 1997; Marcu, 1997, 2000b; Pardo e Rino, 2002; Pardo, 2002); sistemas de auxílio à escrita podem detectar falhas estruturais em textos sendo produzidos (veja, por exemplo, Burstein et al., 2003; Feltrim et al., 2004); tradutores automáticos podem lidar com alguns aspectos organizacionais de textos que são dependentes de língua (veja, por exemplo, Marcu et al., 2000). Entretanto, devido à dificuldade de tratamento desse tipo de informação e à ambigüidade inerente à língua, poucos sistemas são capazes de extrair e manipular conhecimento discursivo. A maior parte dos sistemas atuais limita-se ao uso de ferramentas/recursos mais simples como etiquetadores morfossintáticos (taggers), analisadores sintáticos (parsers) e *wordnets* (isto é, redes de itens lexicais relacionados semanticamente).

Em PLN, a relação entre os níveis de conhecimento comumente distinguidos e a complexidade de manipulação pode ser esquematizada como se mostra na Figura

1.1. Quanto mais se sobe em direção aos níveis da Pragmática e do Discurso, mais complexos são a modelagem e o tratamento computacional. A análise exemplificada anteriormente encontra-se nos níveis mais complexos e abstratos, isto é, nos níveis da Semântica e da Pragmática e do Discurso. Análises desta natureza fazem parte da área de pesquisa conhecida como Análise de Discurso.

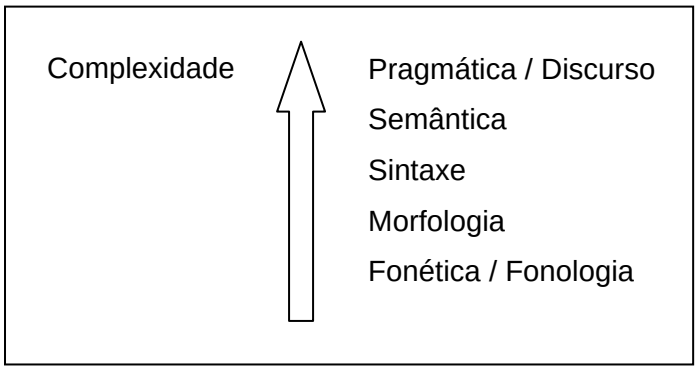


Figura 1.1 – Complexidade e níveis de conhecimento

Um sistema capaz de extrair o conhecimento discursivo subjacente a um texto é chamado analisador discursivo. Sistemas desse tipo podem ser muito úteis e causar um avanço considerável no estado da arte em PLN, principalmente para o português do Brasil, língua muito carente em pesquisas e ferramentas/recursos discursivos.

Para o inglês, há muitas pesquisas sobre discurso e alguns analisadores discursivos já desenvolvidos. Para o português, por outro lado, conhecem-se apenas alguns trabalhos que abordam o tratamento computacional discursivo como é discutido aqui. Rino (1996), por exemplo, define uma modelagem discursiva no contexto da sumarização automática, implementada por Pardo (2002). Roman e Carvalho (2002) apresentam um sistema de diálogo orientado à tarefa com base em um modelo intencional. Feltrim et. al. (2003, 2004) investigam a estruturação esquemática de textos científicos no âmbito de uma ferramenta de auxílio à escrita. Seno e Rino (2005) definem um método de sumarização baseado na organização discursiva de textos para preservação da coerência textual.

Neste contexto, diante da escassez de pesquisas e recursos discursivos para a língua portuguesa, este trabalho de doutorado se faz adequado. Neste trabalho, investigam-se métodos para a análise discursiva automática e produz-se um analisador discursivo automático para o português do Brasil, ferramenta até então inexistente

para esta língua. Seguiu-se a linha de análise discursiva denominada "relacional" (Moore e Pollack, 1992; Moser e Moore, 1996), na qual se estrutura o discurso por meio de relações discursivas entre suas partes. Mais especificamente, a RST foi a principal teoria discursiva adotada para a pesquisa proposta, devido a sua importância e grande utilidade na área.

Pesquisa-se o uso de conhecimentos de naturezas diversas para a automação da análise discursiva. Para isso, duas abordagens são seguidas para o desenvolvimento desta pesquisa: uma simbólica, por meio da explicitação de conhecimento lingüístico para a análise proposta, principalmente, e outra estatística, abordando-se técnicas de Aprendizado de Máquina.

Na linha simbólica, para a análise automática de textos, investiga-se o uso de marcadores discursivos e frases/palavras indicativas da estrutura do discurso, presentes na superfície textual. Tradicionalmente, em PLN, estes marcadores têm sido utilizados para a identificação da estrutura discursiva de textos, pois são fortes sinalizadores dela. Por exemplo, os marcadores "mas" e "portanto" sinalizam, respectivamente, relações de oposição e causa-efeito; a frase indicativa "O objetivo deste trabalho é" no início de uma sentença sinaliza uma possível relação de objetivo; palavras como "positivo", "vantagens" e "satisfatório" em um trecho de texto podem sinalizar a ocorrência de uma relação de avaliação.

O conhecimento necessário para a análise proposta é adquirido por meio da análise de um corpus de textos científicos do domínio da Computação anotado retoricamente segundo a RST, resultando em um repositório de mais de 740 padrões textuais sinalizadores da estrutura discursiva, chamados padrões de análise.

Como resultado mais importante desta pesquisa, produz-se o primeiro analisador discursivo automático para o português do Brasil, chamado DiZer (*DIscourse analyZER*), além de uma grande quantidade de conhecimento discursivo.

Na linha estatística, são propostos três modelos estatísticos inéditos para análise discursiva, baseados no modelo *Noisy-Channel* de Shannon (1948) e treinados pelo método *Expectation-Maximization* (Dempster et al., 1977) com textos reais. Em nível de complexidade crescente, os modelos se baseiam, respectivamente, no conhecimento fornecido por palavras, conceitos e estruturas argumentais para o aprendizado de regras semânticas para a análise automática de textos.

Estes modelos foram desenvolvidos durante realização de um estágio no exterior, na *University of Southern California*, no *Information Sciences Institute*, sob

supervisão do Prof. Dr. Daniel Marcu. Devido à necessidade de ferramentas de alta precisão (como uma *wordnet*, um parser e um sistema reconhecedor de entidades mencionadas) e dados suficientes para o treinamento dos modelos, os modelos foram treinados para a língua inglesa, focando-se a relação causa-efeito. A pesquisa relativa ao modelo baseado em estruturas argumentais, em particular, resultou na produção automática de um repositório de estruturas argumentais para os 1.500 verbos mais freqüentes do inglês, chamado ArgBank.

Todos os métodos de análise propostos são avaliados. O DiZer apresenta desempenho satisfatório, possibilitando seu uso em tarefas de PLN para as quais o conhecimento discursivo que produz possa ser desejável. Os modelos estatísticos mostram-se promissores e devem servir de base para muitos trabalhos derivados deste. O modelo utilizado para a produção do repositório ArgBank produz resultados muito bons, superando diversas limitações de propostas anteriores da literatura. Para a avaliação do DiZer, em particular, produz-se um corpus anotado retoricamente para o qual há plena concordância entre os anotadores. Esse corpus contém textos de vários gêneros e domínios, podendo servir de base para outras pesquisas.

As contribuições deste trabalho para a área de PLN são muitas. São produzidos repositórios de informação discursiva antes inexistentes para o português, que poderão ser explorados com diversas finalidades, podendo subsidiar outras pesquisas. Para o inglês, o repositório de estruturas argumentais ArgBank é produzido. Desenvolve-se o primeiro analisador discursivo para o português e modelos estatísticos são propostos para uma aplicação ainda não modelada desta forma anteriormente, inovando e, portanto, podendo servir de base para pesquisas futuras.

No próximo capítulo, como fundamentação teórica desta pesquisa, apresentam-se a teoria de discurso RST, base deste trabalho de doutorado, e teorias de discurso correlatas consideradas importantes para o completo entendimento do tema tratado nesta pesquisa.

No Capítulo 3, descrevem-se os trabalhos da literatura de maior destaque, em particular, aqueles que desenvolveram analisadores discursivos automáticos para a língua inglesa.

No Capítulo 4, na linha simbólica, relata-se o desenvolvimento do DiZer. São descritas as etapas de montagem, anotação e extração de conhecimento do corpus

utilizado. Apresenta-se a arquitetura do sistema produzido, descrevendo-se seus processos e repositórios de informação. Relata-se, por fim, a avaliação do DiZer.

No Capítulo 5, na linha estatística, os modelos propostos para análise discursiva são descritos. Para isso, introduzem-se o modelo *Noisy-Channel* e o método *Expectation-Maximization*. Descreve-se o corpus utilizado e detalham-se as etapas de treinamento e teste dos modelos. Relata-se, também, o desenvolvimento do repositório de estruturas argumentais, o ArgBank.

Por fim, no Capítulo 6, são apresentadas as conclusões deste trabalho, suas limitações e potencialidades e os trabalhos futuros.

Este trabalho de doutorado foi desenvolvido no NILC[1] (Núcleo Interinstitucional de Lingüística Computacional), um dos maiores grupos de pesquisa em PLN no Brasil, formado por pesquisadores da Universidade de São Paulo (ICMC-USP/São Carlos), da Universidade Federal de São Carlos (UFSCar) e da Universidade Estadual Paulista (UNESP/Araraquara).

[1] http://www.nilc.icmc.usp.br/

2. Teorias Discursivas

Apresentam-se, neste capítulo, a RST – *Rhetorical Structure Theory* (Mann e Thompson, 1987) e teorias discursivas correlatas que pertencem, também, à linha de análise discursiva relacional. Além disso, abordam-se algumas pesquisas que tentaram relacionar os níveis de representação abordados por cada teoria, estabelecendo formas de mapeamento entre elas.

2.1. Rhetorical Structure Theory

Segundo Hovy (1988), a retórica é a parte "palpável" da pragmática, através da qual se estabelece a coerência de um texto. Ela é o meio pelo qual um texto é organizado para satisfazer seu objetivo comunicativo subjacente, representando, portanto, a organização funcional do texto, ou seja, qual a função de suas partes para que o objetivo comunicativo do texto seja satisfeito.

Como especificado no capítulo anterior, a *Rhetorical Structure Theory* – RST (Mann e Thompson, 1987) é a principal teoria discursiva adotada neste trabalho. Apesar de ter sido desenvolvida para análise de textos, ela vem sendo utilizada para os mais diversos fins em PLN. Por estas razões, justifica-se sua escolha para o desenvolvimento desta pesquisa.

Na RST, são definidas 26 relações retóricas que são utilizadas para relacionar as proposições expressas em um texto, construindo sua estrutura retórica. As proposições relacionadas podem ser simples, se expressarem um único fato ou evento, ou, caso contrário, complexas. Segundo os autores da teoria, se um texto for coerente, será possível construir sua estrutura retórica. Por este motivo, as relações retóricas também são chamadas relações de coerência.

Em casos padrões, as relações se estabelecem entre duas proposições, normalmente (mas não necessariamente) expressas por segmentos adjacentes no texto, sendo uma nuclear (N) e outra complementar (S – "satélite"), indicando, respectivamente, a informação principal para a satisfação da intenção subjacente à relação e uma informação adicional, a qual influencia de alguma forma a interpretação que o leitor faz da informação nuclear. Quando ambas as informações relacionadas são igualmente importantes, diz-se que se tem uma relação multinuclear, isto é, com

mais de um núcleo e nenhum satélite. Por exemplo, no trecho de texto “Embora você não goste, trabalhar é importante.”, a proposição expressa pela primeira oração é o satélite e a proposição expressa pela segunda é o núcleo da relação retórica de oposição CONCESSION. Por sua vez, no trecho de texto “O garoto chegou da escola e fez sua lição de casa. Depois, foi brincar com os amigos.”, há uma relação SEQUENCE (indicando uma “seqüência” de eventos) entre as proposições expressas pelas orações “O garoto chegou da escola”, “e fez sua lição de casa.” e “Depois, foi brincar com os amigos.”, sendo que todas são consideradas núcleos da relação, pois possuem a mesma importância. Nesses casos, CONCESSION é uma relação mononuclear, enquanto SEQUENCE é multinuclear.

As relações definidas na RST são mostradas na primeira coluna da Tabela 2.1 (em inglês, como na obra de referência). Segundo seus autores, estas relações podem ser aplicadas a uma grande gama de textos sem necessitarem de alterações. Na segunda coluna da tabela, as relações multinucleares são identificadas.

Tabela 2.1 – Relações retóricas da RST (Mann e Thompson, 1987)

Relações	Multinuclear	Natureza das relações
ANTITHESIS	Não	Intencional
BACKGROUND	Não	Intencional
CIRCUMSTANCE	Não	Semântica
CONCESSION	Não	Intencional
CONDITION	Não	Semântica
ELABORATION	Não	Semântica
ENABLEMENT	Não	Intencional
EVALUATION	Não	Semântica
EVIDENCE	Não	Intencional
INTERPRETATION	Não	Semântica
JUSTIFY	Não	Intencional
MEANS	Não	Semântica
MOTIVATION	Não	Intencional
NON-VOLITIONAL CAUSE	Não	Semântica
NON-VOLITIONAL RESULT	Não	Semântica
OTHERWISE	Não	Semântica
PURPOSE	Não	Semântica
RESTATEMENT	Não	Semântica
SOLUTIONHOOD	Não	Semântica
SUMMARY	Não	Semântica
VOLITIONAL CAUSE	Não	Semântica
VOLITIONAL RESULT	Não	Semântica
CONTRAST	Sim	Semântica
JOINT	Sim	Semântica
LIST	Sim	Semântica
SEQUENCE	Sim	Semântica

A definição de cada relação especifica quatro tipos de informação necessários para determinar sua ocorrência entre duas proposições, quer na produção do texto pelo escritor, quer na sua interpretação por um leitor. Essas informações são:

- restrições sobre o núcleo (N);
- restrições sobre o satélite (S);
- restrições sobre a combinação do núcleo e do satélite (N+S);
- efeito pretendido (ou intenção): especificação do efeito que a relação em questão causa no leitor, quando este interpreta um texto, ou do efeito pretendido pelo escritor, quando este seleciona tal relação para estruturar seu texto.

Como exemplo, considere as definições abaixo para as relações CONCESSION e JUSTIFY.

Nome da relação: CONCESSION
Restrições sobre N: o escritor julga N válido **Restrições sobre S**: o escritor não afirma que S pode não ser válido **Restrições sobre N+S**: o escritor mostra uma incompatibilidade aparente ou em potencial entre N e S; o reconhecimento da compatibilidade entre N e S melhora a aceitação de N pelo leitor **Efeito**: o leitor aceita melhor N

Figura 2.1 – Definição da relação retórica CONCESSION

Nome da relação: JUSTIFY
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: a compreensão de S pelo leitor aumenta sua prontidão para aceitar o direito do escritor de apresentar N **Efeito**: a prontidão do leitor para aceitar o direito do escritor de apresentar N aumenta

Figura 2.2 – Definição da relação retórica JUSTIFY

Com essas duas relações, é possível construir a estrutura retórica correspondente ao trecho de texto “Embora você não goste, trabalhar é importante. O trabalho enobrece o homem.”, como se mostra na Figura 2.3.

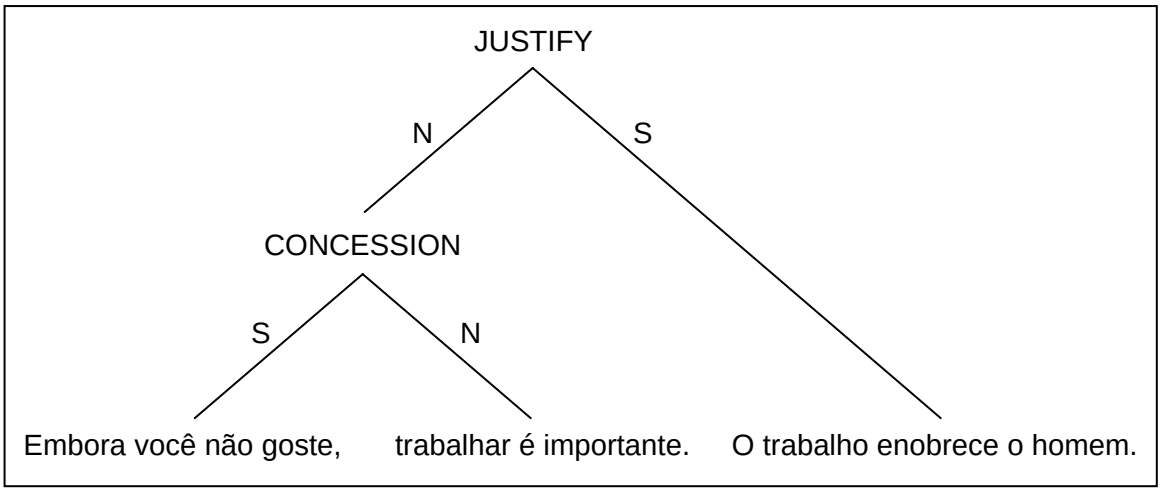


Figura 2.3 – Exemplo de estrutura retórica

É comum, neste nível de análise, que haja mais de uma estrutura retórica possível para um texto, como os próprios autores da RST reconhecem. O principal motivo para isso é que a estrutura discursiva subjacente a um texto depende da interpretação deste. Esta interpretação, por sua vez, é puramente subjetiva, podendo variar com o leitor do texto. Outro motivo é que a RST não possui uma formalização robusta (Marcu, 2000b), fato atribuído à dificuldade de se escolher uma única relação retórica para relacionar duas proposições com base somente nas definições das relações fornecidas.

As variações nas estruturas retóricas de um texto podem ocorrer em vários níveis: variação no que se considera como informação nuclear ou não no texto; variação nas relações retóricas escolhidas para relacionar as proposições; variações na estruturação retórica em si, produzindo-se estruturas com diferentes formas. Exemplos dessas variações na estrutura da Figura 2.3, supondo-se diferentes situações nas quais o texto é produzido, poderiam ser: a justificativa "O trabalho enobrece o homem" poderia ser considerada mais importante do que o primeiro trecho de texto, devendo, portanto, ser considerada núcleo de alguma relação, em vez de satélite; em vez de uma relação CONCESSION entre as duas primeiras proposições, poderia haver uma relação ANTITHESIS, a qual também pertence ao grupo das relações de oposição; a relação JUSTIFY poderia ser estabelecida entre a terceira e segunda proposição apenas, em vez das duas primeiras proposições, alterando, desta maneira, a forma da estrutura retórica.

Uma distinção importante de se observar na RST refere-se à natureza das relações. Como comentado no capítulo anterior, a RST possui tanto relações retóricas

de natureza intencional/argumentativa quanto semântica/informativa/factual. A terceira coluna da Tabela 2.1 identifica a natureza de cada relação. Muito se discute na literatura sobre essa distinção. Alguns autores consideram que o nível retórico de análise deveria conter somente as relações intencionais (veja, por exemplo, Jordan, 1992; Moser e Moore, 1996). Outros afirmam que a ambigüidade que ocorre neste nível de análise deve-se ao fato de a RST considerar os dois tipos de relação, pois é possível que duas relações, uma intencional e outra semântica, sejam escolhidas para relacionar duas proposições quaisquer (veja, por exemplo, Hovy, 1991; Moore e Pollack, 1992; Koreslky e Kittredge, 1993; Moser e Moore, 1996). Moore e Pollack ilustram esse problema com o trecho de texto abaixo (em inglês):

(a) *George Bush supports big business.* (b) *He's sure to veto House Bill 1711.*

no qual é possível se estabelecer uma relação retórica EVIDENCE, de natureza intencional, ou uma relação retórica VOLITIONAL CAUSE, de natureza semântica, nas quais a proposição correspondente ao segmento (b) é a informação nuclear. Com estas possibilidades, tem-se que (a) evidencia (b) ou que (a) causa (b). Nas Figuras 2.4 e 2.5, mostram-se as definições destas duas relações.

Nome da relação: EVIDENCE
Restrições sobre N: o leitor pode não acreditar em N de forma satisfatória para o escritor **Restrições sobre S**: o leitor acredita em S ou o acha válido **Restrições sobre N+S**: a compreensão de S pelo leitor aumenta sua crença em N **Efeito**: a crença do leitor em N aumenta

Figura 2.4 – Definição da relação retórica EVIDENCE

Nome da relação: VOLITIONAL CAUSE
Restrições sobre N: apresenta uma ação volitiva ou uma situação que poderia surgir de uma ação volitiva **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta uma situação que pode ter acarretado o fato do agente da ação volitiva em N ter realizado a ação; sem S, o leitor poderia não reconhecer a motivação da ação; N é mais central para a satisfação do objetivo do escritor do que S **Efeito**: o leitor reconhece que a situação apresentada em S como a causa da ação apresentada em N

Figura 2.5 – Definição da relação retórica VOLITIONAL CAUSE

Os autores da RST afirmam que sempre haverá uma relação considerada mais proeminente em um determinado contexto. Jordan (1992), por sua vez, declara que é "ingenuidade" tentar escolher apenas uma relação entre duas proposições.

Neste trabalho de doutorado, consideram-se tanto as relações retóricas intencionais quanto as semânticas, assumindo, como os autores da RST, que, diante de ambigüidades, haverá uma relação mais proeminente. No analisador discursivo desenvolvido neste trabalho, em particular, há um repositório de dados estatísticos que permite o ranqueamento das estruturas retóricas produzidas e, desta forma, possibilita, quando desejável, a escolha de uma organização discursiva entre várias possíveis.

Vários autores modificam, complementam ou discutem a especificação da RST e de suas relações para diversos fins, tanto lingüísticos quanto computacionais. Marcu (1997, 2000b), em especial, adiciona ao conjunto de relações retóricas da RST as relações chamadas estruturais, sem significado aparente, mas que auxiliam na estruturação retórica de textos. Exemplos destas relações são as relações PARENTHETICAL e SAME-UNIT. A relação PARENTHETICAL indica informação extra, que não pertence ao corpo principal do texto (por exemplo, informação entre parênteses, colchetes e chaves ou especificada como nota de rodapé). A relação SAME-UNIT é utilizada para unir segmentos textuais não adjacentes no texto que expressam uma única proposição. Esse tipo de estruturação discursiva ocorre, por exemplo, quando há uma oração relativa no interior de uma sentença. Essa oração, além de usualmente expressar uma proposição por si só, separa segmentos que também expressam uma única proposição. O mesmo acontece quando há uma relação PARENTHETICAL no interior de uma sentença. Na Figura 2.6, mostra-se um trecho de texto estruturado retoricamente no qual se podem observar as duas relações estruturais anteriores.

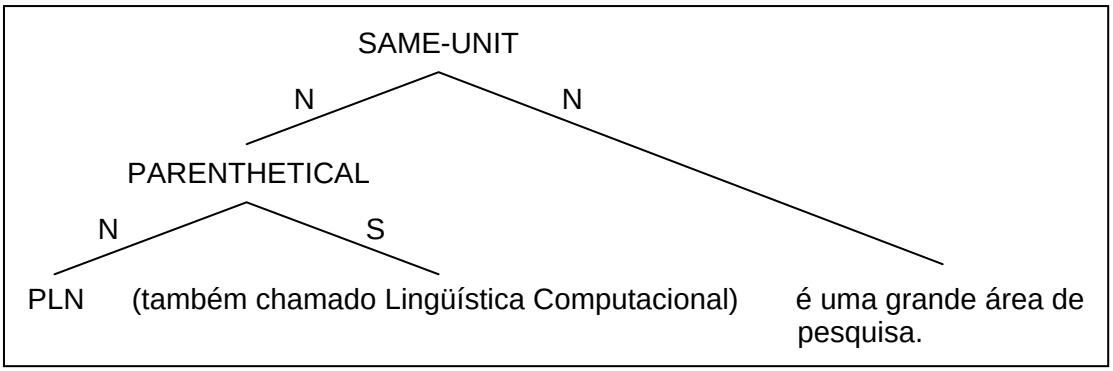


Figura 2.6 – Exemplo de relações PARENTHETICAL e SAME-UNIT

A relação SAME-UNIT é sempre caracterizada como multinuclear, pois conecta segmentos que formarão uma única proposição, não havendo mais de uma proposição para que satélite e núcleo sejam determinados. A relação PARENTHETICAL por sua vez, é sempre caracterizada como mononuclear, com a informação extra sendo o satélite da relação. Isso é explicado pelo fato de que, em nível discursivo, a proposição expressa pelo trecho que contém a informação extra é menos importante no contexto em que ocorre. Neste trabalho de doutorado, as duas relações estruturais apresentadas são utilizadas.

A maioria dos analisadores discursivos automáticos existentes baseia-se na RST. Alguns estendem seu conjunto de relações, enquanto outros simplificam, adotando conjuntos de relações mais genéricos.

No Apêndice A, todas as relações retóricas utilizadas neste trabalho de doutorado são definidas. Estas relações, em sua maioria, são as relações originais da RST.

2.2. Relações Intencionais de Grosz e Sidner (1986)

Segundo Grosz e Sidner (1986), todo discurso é essencialmente produzido com a finalidade de satisfazer uma ou mais intenções. São as intenções que individualizam e tornam coerente o discurso.

De acordo com a teoria proposta pelas autoras, referenciada como *Grosz and Sidner Discourse Theory* (GSDT), quando um escritor escreve seu texto, ele produz e estrutura o conteúdo do texto em função de suas intenções. Nestes termos, há dois tipos de intenções: a intenção primária do discurso e as intenções subjacentes aos

segmentos do discurso, as quais devem contribuir para a satisfação da intenção primária. Por exemplo, para o texto (já segmentado) exibido na Figura 2.7 abaixo, a intenção primária pode ser "convencer o leitor de que autômatos finitos são excelentes estruturas para a representação de grandes dicionários de língua natural". Por sua vez, a intenção subjacente ao segmento 1 pode ser "fazer com que o leitor saiba que a representação de grandes dicionários de língua natural é um interessante problema computacional a ser tratado", que, assim como as intenções subjacentes aos outros segmentos, contribui para a satisfação da intenção primária. É o reconhecimento destas intenções que permite ao leitor recuperar o que o escritor pretendia comunicar com o texto.

[A representação de grandes dicionários de língua natural, principalmente nos casos em que se trabalha com vários milhões (ou dezenas de milhões) de palavras, é um interessante problema computacional a ser tratado dentro da área de Processamento de Língua Natural.]$_1$ [Autômatos finitos, largamente usados na construção de compiladores, são excelentes estruturas para representação desses dicionários,]$_2$ [permitindo acesso direto aos às palavras e seus possíveis atributos.]$_3$ [Um dicionário contendo mais de 430.000 palavras da língua portuguesa sem atributos, cuja representação em formato texto ocupa mais de 4.5Mb, pode ser convertido em um autômato compactado de apenas 218Kb.]$_4$

Figura 2.7 – Texto-exemplo

Como as intenções possíveis em um discurso são infinitas (tanto as primárias quanto as subjacentes aos segmentos), como as próprias autoras afirmam, a teoria proposta organiza o discurso por meio de relações de contribuição e satisfação entre as intenções, que, por sua vez, são finitas. Estas relações, chamadas de relações intencionais, são duas:

- *dominance* (DOM): se a intenção subjacente a um segmento Y contribui para a satisfação da intenção subjacente ao segmento X, então se diz que a intenção subjacente a X domina (*dominates*) a intenção subjacente a Y, ou seja, DOM(X,Y);
- *satisfaction-precedence* (SP): se a intenção subjacente a um segmento X deve ser satisfeita antes da intenção subjacente a um segmento Y, então se diz que a satisfação da intenção subjacente a X deve preceder (*satisfaction-precedes*) a satisfação da intenção subjacente a Y, ou seja, SP(X,Y).

Como exemplo, para o texto da Figura 2.7, as seguintes relações podem ser verificadas: DOM([2-4],1), DOM([2-3],4) e DOM(2,3), nas quais [X-Y] indica o segmento discursivo formado pelos segmentos textuais de X a Y. Não há, neste discurso, nenhuma relação SP "significativa" (nos termos da teoria), as quais são muito comuns em discursos que apresentam seqüências de eventos.

A teoria também define duas relações informativas que não se estabelecem entre intenções, mas entre o conteúdo proposicional dos segmentos. Elas são:

- *supports* (SUP): se a crença do leitor no conteúdo de um segmento Y fornece subsídios para que o leitor creia no conteúdo de um segmento X, então se diz que o conteúdo de Y sustenta (*supports*) o conteúdo de X, ou seja, SUP(Y,X);
- *generates* (GEN): se a execução de uma ação descrita no segmento Y contribui para a execução de uma ação descrita no segmento X, então se diz que Y gera (*generates*) X, ou seja, GEN(Y,X).

Grosz e Sidner determinam, ainda, uma correspondência direta entre a relação DOM e as relações SUP e GEN. Para X e Y quaisquer, tem-se as seguintes correspondências:

- no caso de crença, DOM(X,Y) $\Leftrightarrow$ SUP(Y,X);
- no caso de ação, DOM(X,Y) $\Leftrightarrow$ GEN(Y,X).

nas quais o símbolo $\Leftrightarrow$ representa uma implicação de dois sentidos. Em outras palavras, caso se verifique que o discurso trata de crenças (ações), é possível determinar DOM em função de SUP (GEN), assim como SUP (GEN) em função de DOM.

Há uma forte correlação entre a GSDT e a RST. As relações retóricas da RST são determinadas na análise de um texto em função das intenções percebidas, as quais são especificadas no campo "Efeito" das definições das relações. Entretanto, tais intenções não são feitas explícitas neste campo, dado o fato de que há infinitas intenções. Na GSDT, modelam-se justamente as intenções do discurso em termos das relações intencionais. A GSDT complementa a RST neste aspecto.

2.3. Relações Semânticas de Jordan (1992)

De acordo com Jordan (1992), uma relação semântica constitui uma "noção semântica textual de conexão binária entre quaisquer duas partes de um texto".

As relações semânticas de Jordan são, na realidade, um amálgama das relações propostas em vários outros trabalhos importantes em PLN, destacando-se os trabalhos de Winter (1968, 1971, 1974, 1976, 1977, 1979, 1982), Hoey (1979, 1983a, 1983b), Hoey e Winter (1986) e do próprio Jordan (1978, 1980, 1984, 1985a, 1985b, 1988, 1989). A lista completa de relações semânticas propostas por Jordan é mostrada na primeira coluna da Tabela 2.2. A segunda coluna indica a tipologia das relações definida pelo próprio autor.

Tabela 2.2 – Relações semânticas de Jordan (1992)

Relações	Tipo das Relações
Identification	*Detail*
Classification	
Specification	
Appearance	
Characteristics	
Function	
Material	
Parts	
Active	*General*
Passive	
Agent	
Source	
Assessment	*Logical*
Basis	
Cause	
Effect	
Emotive Effect	
Purpose	
Means	
Problem	
Solution	
Possibility	*Modal*
Capability	
Correctness	
Propriety	
Necessity	
Need	
Completion	
Achievement	
Future	
Intention	
Mandate	

Authority	
Determination	
Permission	
Obligation	
Willingness	
Desire	
Time	*Time*
Before	
After	
Simultaneous	
Inverted time	
Elaboration	*Text manipulation*
Summary	
Repetition	
Paraphrase	
Forecast	
Transition	
Collateral inversion	*Special*
Concession	
Compatibility	
Contrast	
Comparison	
Conditionals	
Document structures	
Hypothetical-Real	
Transition couplets	
Accompaniment	*Other*
Circumstance	
Inverted circumstance	
Connection	
Enablement	
Example	
Extent	
Location	
Inverted Location	
Manner	
True	

Segundo Jordan, estas relações capturam a forma como os conhecimentos contidos em um texto se relacionam, sendo completamente desvinculadas das intenções do escritor. Sob a visão de Jordan, esta característica das relações semânticas é o que as diferenciam das relações retóricas, principalmente das relações retóricas de natureza semântica da RST. Neste caso específico, apesar das relações da RST e das relações de Jordan estabelecerem relações entre os conteúdos proposicionais de trechos de textos, não interferindo nas inclinações pessoais do leitor, as relações da RST identificam o que é nuclear ou não para a satisfação do objetivo comunicativo

pretendido (a intenção subjacente), enquanto as relações semânticas "puras" de Jordan, não (como sugerem, também, Moser e Moore, 1996). Por exemplo, para o trecho de texto "Um incêndio destruiu várias casas. Algumas pessoas foram para o hospital.", tanto a relação retórica NON-VOLITIONAL CAUSE quanto a relação NON-VOLITIONAL RESULT poderiam ser utilizadas para relacionar as duas sentenças: se o trecho mais importante para a satisfação do objetivo comunicativo do escritor do texto for o primeiro (que se refere ao incêndio), a relação NON-VOLITIONAL RESULT deve ser usada; caso contrário, se o trecho mais importante para a satisfação do objetivo comunicativo do escritor for o segundo (que se refere às pessoas que foram para o hospital), a relação NON-VOLITIONAL CAUSE deve ser usada. As relações semânticas de Jordan, por sua vez, apenas estabelecem a relação existente entre dois conteúdos proposicionais, não tendo a função de indicar o que é mais importante ou não para qualquer que seja o objetivo comunicativo pretendido pelo escritor do texto. Para o exemplo anterior, haveria apenas uma relação semântica de causa-efeito entre as proposições, indicando que o incêndio causou o fato de algumas pessoas terem ido para o hospital, não atribuindo, assim, maior importância a nenhuma delas.

Assim como na RST, entre os problemas encontrados na determinação das relações semânticas em um texto, há a ambigüidade inerente a este nível de análise, ou seja, quando mais de uma relação é possível entre duas proposições. É interessante notar, neste contexto, o grande elenco de relações da Tabela 2.2 e a tênue diferença entre algumas delas (por exemplo, *elaboration* e *example*; *willingness* e *desire*; *classification* e *specification*).

2.4. Relações Semânticas de Kehler (2002)

Diferentemente de Jordan, Kehler (2002) define apenas três relações semânticas: *resemblance*, *cause-effect* e *contiguity*. Kehler afirma que estas três relações são suficientes para estruturar qualquer discurso e, similarmente às relações das outras teorias discursivas, podem ser utilizadas para lidar com uma grande gama de questões em PLN.

Segundo Kehler, as três relações podem ser diferenciadas de acordo com as proposições que relacionam e com os tipos de inferência necessários para que se

identifiquem suas aplicações: a relação *resemblance* requer que haja coisas em comum e/ou em contraste entre as entidades das proposições relacionadas; a relação *cause-effect* requer que seja possível perceber uma implicação, direta ou indireta, conectando as proposições relacionadas; *contiguity*, por fim, requer que uma seqüência de eventos envolvendo as entidades das proposições relacionadas seja expressa.

Segundo o autor, essas relações são básicas e todas as relações discursivas de outras teorias podem ser classificadas como uma destas três relações. Por exemplo, as relações ELABORATION, LIST e CONTRAST da RST são, de acordo com suas definições, relações *resemblance*; as relações CAUSE e RESULT (volitivas ou não), EXPLANATION e JUSTIFY da RST são relações *cause-effect*; a relação SEQUENCE da RST é uma relação *contiguity*.

Neste trabalho, na abordagem estatística investigada, a relação semântica *cause-effect* de Kehler é a relação que mais se aproxima da relação tratada pelos modelos propostos.

2.5. Mapeamento entre Relações do Discurso

Conforme as seções anteriores, é possível analisar um discurso em vários níveis, com diferentes perspectivas e objetivos em mente. Na literatura, muito se tem discutido como tais níveis podem co-existir, como um nível influencia o outro e se há ou não mapeamentos possíveis entre eles. A seguir, são relatadas as principais pesquisas que estabeleceram possíveis relacionamentos e mapeamentos entre os níveis discursivos. Na Subseção 2.5.1, mostra-se a relação entre a retórica e as intenções, enquanto, na Subseção 2.5.2, mostra-se a relação entre a retórica e a semântica, discutindo-se os problemas encontrados e as soluções sugeridas na literatura.

2.5.1. Retórica e Intenções

Pesquisas recentes na área de Análise de Discurso têm mostrado e concordado com o fato de que a retórica é a forma de expressão das intenções no discurso (Maier e Hovy, 1991; Maybury, 1992; Dale, 1993; Hovy, 1993; Maier, 1993; Moore e Paris, 1993; Moore, 1995; Moser e Moore, 1996; Rino, 1996; Marcu, 1999, 2000a; Pardo,

2002; etc.). Assim, quando um escritor produz um texto, ele tem em mente um objetivo comunicativo, uma intenção, a atingir. Por meio das relações retóricas, ele organiza e estrutura o conteúdo textual de forma que sua intenção seja satisfeita. Ressalta-se, porém, que este mapeamento de intenções em relações retóricas não é simples, pois (a) uma mesma intenção pode ser realizada por diferentes estratégias retóricas e (b) uma mesma estratégia retórica pode servir para a realização de diferentes intenções. Como exemplo desse relacionamento complexo, na Tabela 2.3, mostra-se parte do mapeamento identificado por Moore e Paris entre intenções e relações retóricas para a aplicação que desenvolveram, a qual consiste em um sistema de diálogo para gerar explicações. Por exemplo, a intenção de "capacitar o leitor a identificar algo" (segunda linha da tabela) pode ser realizada pelas relações retóricas CIRCUMSTANCE, CONDITION e CONTRAST, entre outras. Por outro lado, a relação retórica CONTRAST pode ser a realização da intenção de "capacitar o leitor a identificar algo" e "fazer com que o leitor acredite em uma proposição".

Tabela 2.3 – Mapeamento de intenções em relações retóricas de Moore e Paris (1993)

Intenções	Relações Retóricas
▪ persuadir o leitor sobre uma proposição ▪ persuadir o leitor a realizar uma ação ▪ tornar o leitor competente para compreender algo ▪ tornar o leitor competente para realizar uma ação	EVIDENCE MOTIVATION BACKGROUND ENABLEMENT
▪ capacitar o leitor a identificar algo	CIRCUMSTANCE CONDITION CONTRAST ELABORATION PURPOSE SEQUENCE
▪ fazer com que o leitor acredite em uma proposição	CONTRAST ELABORATION

Por serem as teorias discursivas mais representativas sobre intenções e retórica, respectivamente, a GSDT e a RST têm sido objetos de estudo de muitas pesquisas. As relações da RST, em especial, têm implícitas no campo "Efeito" de suas definições as intenções que pretendem atingir. Por exemplo, a relação CONCESSION pretende que o leitor aumente sua convicção na proposição nuclear, qualquer que seja ela. A GSDT, reconhecendo a infinidade de intenções existentes, propôs a estruturação discursiva com base na relação entre as intenções do discurso.

Alguns trabalhos tentaram unir as duas teorias e seus pressupostos teóricos. Apesar das diferenças evidentes entre a RST e a GSDT, que vão desde o que se considera como unidade elementar de análise (proposições vs. um ou mais segmentos textuais que satisfazem uma intenção) até a própria estruturação (retórica vs. intencional), similaridades foram encontradas, permitindo o mapeamento (às vezes parcial) de um nível no outro.

Moser e Moore (1996) foram as primeiras a reconhecer a correspondência entre os conceitos de nuclearidade da RST e de dominância da GSDT. Elas verificaram que, em uma relação retórica padrão, isto é, com um núcleo e um satélite, a intenção subjacente ao núcleo domina (DOM) a intenção subjacente ao satélite. O inverso também é possível, ou seja, quando um segmento domina outro, pode-se dizer que o primeiro será o núcleo de uma relação retórica e o segundo o satélite.

No caso de relações multinucleares, em que não há um satélite para ser dominado pelo(s) núcleo(s), Moser e Moore propõem a hipótese de que, talvez, a RST e a GSDT possuam pressupostos teóricos incompatíveis neste ponto. Justamente nesta questão, Marcu (1999) estendeu o trabalho de Moser e Moore afirmando que, em relações multinucleares, não há dominância entre os segmentos, mas pode haver precedência (SP). Posteriormente, Marcu (2000a) formalizou esse mapeamento entre relações retóricas e intencionais. Além disso, mostrou que é possível derivar a intenção primária de um discurso por meio de sua estrutura retórica. Segundo ele, a intenção primária é dada pela proposição mais nuclear da estrutura retórica em conjunto com a relação retórica correspondente. Na estrutura do texto da Figura 2.7, mostrada a seguir na Figura 2.8, por exemplo, a intenção primária seria dada pela combinação da proposição expressa pelo segmento 2, que é a mais nuclear da estrutura, com a relação retórica EVIDENCE, resultando na intenção primária "aumentar a convicção do leitor de que autômatos finitos são excelentes estruturas para a representação de dicionários de língua natural", na qual o fato de "aumentar a convicção do leitor em algo" provém do campo "Efeito" da definição da relação EVIDENCE, enquanto o fato de que "autômatos finitos são excelentes estruturas para a representação de dicionários de língua natural" provém da proposição expressa pelo segmento 2.

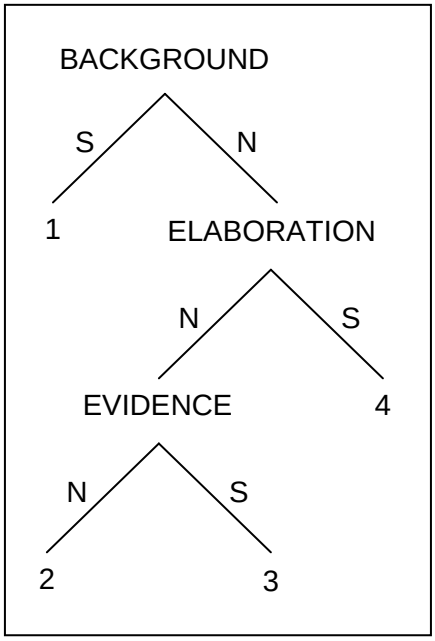


Figura 2.8 – Estrutura retórica para o texto da Figura 2.7

Da mesma forma que as relações intencionais podem ser determinadas a partir da estrutura retórica, Marcu enfatiza que as relações intencionais também podem ser usadas para restringir as possíveis estruturas retóricas de um texto em um processo automático de análise, caso estas relações estejam disponíveis antes da análise, o que, normalmente, não se observa em problemas reais de PLN.

Moser e Moore e também Marcu partiram dos pressupostos teóricos da RST e da GSDT para determinar o relacionamento entre retórica e intenções. Em uma linha oposta (empírica), Rino (1996), por meio de análise de corpora, estabeleceu um mapeamento de relações semânticas e intencionais em relações retóricas. Esse mapeamento (aprimorado por Pardo, 2002) é mostrado na Tabela 2.4. Assume-se, na coluna das relações retóricas, que o primeiro argumento das relações retóricas é o núcleo e o segundo é o satélite, excetuando-se os casos de relações multinucleares (LIST, CONTRAST e SEQUENCE), em que os dois argumentos são núcleos da relação. O mapeamento funciona da seguinte forma: dada uma relação semântica entre duas proposições X e Y em uma base de conhecimento qualquer e as relações intencionais entre estas proposições, é possível produzir uma relação retórica envolvendo as duas proposições. De forma inversa, dada uma relação retórica, é possível desmembrá-la em uma relação semântica e algumas relações intencionais.

Tabela 2.4 – Mapeamento de Rino (1996)

Caso	Relações semânticas	Relações retóricas	Relações intencionais
1	*enable*(Y,X)	PURPOSE(Y,X) MEANS(X,Y)	SP(X,Y) DOM(Y,X)
2	*rationale*(Y,X)	PURPOSE(X,Y) JUSTIFY(X,Y)	SUP(Y,X) DOM(X,Y) not SP(X,Y)
3	*proof*(X,Y)	EVIDENCE(X,Y) JUSTIFY(X,Y)	SP(Y,X) DOM(X,Y) SUP(Y,X)
4	*cause*(X,Y)	RESULT(Y,X) CAUSE(X,Y)	SP(Y,X) DOM(X,Y) GEN(Y,X)
5	*simSem*(X,Y)	LIST(X,Y)	DOM(X,Y) DOM(Y,X)
6	*difSem*(X,Y)	CONTRAST(X,Y)	DOM(X,Y) DOM(Y,X)
7	*attribute*(X,Y) *detail*(X,Y) *exemplify*(X,Y)	ELABORATION(X,Y)	SUP(Y,X)
8	*evalSem*(X,Y)	EVALUATION(X,Y)	DOM(X,Y)
9	*reason*(X,Y)	EXPLANATION(X,Y)	GEN(Y,X)
10	*sequence*(X,Y)	SEQUENCE(X,Y)	SP(X,Y)
11	*backSem*(X,Y)	BACKGROUND(X,Y)	not SP(X,Y) SUP(Y,X)

O mapeamento de Rino distingue as relações retóricas das semânticas pela força argumentativa das relações: a retórica tem força argumentativa, enquanto a semântica não. Segundo Rino, a força argumentativa das relações retóricas é dada pelas relações intencionais que se estabelecem entre as proposições. De fato, Pardo (2002) constata que são as relações intencionais que atribuem às proposições o caráter nuclear ou não nuclear durante a construção das estruturas retóricas.

2.5.2. Retórica e Semântica

O mapeamento entre retórica e semântica não foi tão explorado e formalizado quanto o mapeamento entre retórica e intenções. Alguns trabalhos, como Moser e Moore (1996), Moore e Pollack (1992) e Hovy (1991, 1993), sugerem que um texto deve ter as relações semânticas incorporadas à sua estrutura retórica, mas não explicitam como as relações retóricas e semânticas se relacionam no discurso.

Korelsky e Kittredge (1993) sugerem que as relações retóricas se estabelecem entre proposições relacionadas semanticamente. Eles ressaltam que, como acontece no mapeamento entre intenções e retórica, uma relação retórica pode se servir de várias relações semânticas no discurso, assim como uma relação semântica pode ser interpretada como várias relações retóricas. Por exemplo, uma relação retórica EVIDENCE pode ser observada entre proposições conectadas pelas relações semânticas VOLITIONAL CAUSE (trecho de texto 1), NON-VOLITIONAL CAUSE (trecho de texto 2) e ELABORATION (trecho de texto 3), como mostram os trechos de texto abaixo retirados integralmente dos trabalhos de Korelsky e Kittredge (trechos 1 e 2) e de Mann e Thompson (1987) (trecho 3).

(1) *George Bush supports big business.*
He's sure to veto House Bill 1711.

(2) *Winters in Montreal are so cold.*
I need a fur coat.

(3) *George Bush definitely supports big business.*
He just voted House Bill 1711.

Para exemplificar o caso contrário, Korelsky e Kittredge usam o trecho de texto 4 abaixo, no qual há uma relação semântica CONDITION, podendo-se reconhecer tanto a relação retórica ENABLEMENTE quanto a relação MOTIVATION.

(4) *Come home by 5:00.*
Then we can go to the hardware store before it closes.

Como se pode notar, as relações semânticas citadas nos exemplos anteriores pelos autores são as relações retóricas de natureza semântica da RST.

Korelsky e Kittredge sugerem algoritmos para determinar a relação semântica a partir da retórica. Para o caso da relação retórica EVIDENCE e suas correspondentes semânticas, o algoritmo da Figura 2.9 é dado como exemplo.

Se a relação retórica EVIDENCE é observada entre duas proposições P1 e P2, em que P1 é o núcleo e P2 é o satélite, então:

1) se há um agente consciente de tal forma que P1 e P2 fazem referência a suas ações, então a relação semântica VOLITIONAL CAUSE se estabelece entre as proposições;
2) se não há um agente consciente, então a relação semântica NON-VOLITIONAL CAUSE se estabelece entre as proposições;
3) se P2 é uma proposição genérica, então a relação semântica ELABORATION se estabelece entre as proposições.

Figura 2.9 – Algoritmo de Korelsky e Kittredge (1993) para mapeamento da relação retórica EVIDENCE em possíveis relações semânticas

Na mesma linha de Korelsky e Kittredge, Hovy (1991) sugere que as próprias definições das relações retóricas sejam enriquecidas com as informações semânticas. Entretanto, essa mudança na definição das relações retóricas causaria vários problemas (como apontado por Moore e Pollack): perda de modularidade das análises retórica e semântica no tratamento discursivo; proliferação das definições das relações, já que uma relação retórica pode ser observada com a ocorrência de várias relações semânticas; e, mais importante, as estruturas retórica e semântica podem não ser isomórficas, ou seja, podem possuir formatos estruturais diferentes.

De fato, Dale (1993) sugere que, diferentemente da retórica, a estrutura semântica de um texto é, normalmente, um grafo. Teoricamente, mais de uma relação semântica pode se estabelecer entre as proposições expressas por quaisquer dois segmentos do texto. Para resolver esse problema, Moser e Moore sugerem que as relações semânticas sejam "parasitas" das relações retóricas, isto é, que elas se estabeleçam somente entre as proposições relacionadas pelas relações retóricas em questão (como fazem Korelsky e Kittredge), o que faria com que as estruturas retóricas e semânticas se tornassem isomórficas.

Neste trabalho de doutorado, como já mencionado, o analisador discursivo simbólico desenvolvido, o DiZer, baseia-se na RST. Na abordagem estatística, os modelos desenvolvidos foram treinados para o reconhecimento de relações semânticas causa-efeito, similares à relação *cause-effect* de Kehler, como será discutido posteriormente.

Diante das possibilidades de mapeamento entre as relações do discurso, faz-se possível, por exemplo, a derivação imediata das relações intencionais da GSDT a partir da estrutura retórica produzida pelo DiZer para um texto. De forma similar,

pode-se mapear as relações retóricas detectadas pelo sistema para as relações semânticas de Kehler. Para mapeamentos mais sofisticados, envolvendo o conjunto de relações semânticas de Jordan, por exemplo, acredita-se na necessidade de algoritmos como os sugeridos por Korelsky e Kittredge (1993). Essas possibilidades serão discutidas no Capítulo 6.

Para mais detalhes sobre as teorias discursivas discutidas anteriormente, assim como sobre o mapeamento entre elas, veja Pardo e Nunes (2003a).

No próximo capítulo, os aspectos dos trabalhos da literatura sobre análise discursiva automática considerados importantes para esta pesquisa são descritos.

3. Trabalhos Correlatos

Recentemente, em PLN, vários trabalhos têm apresentado modelos formais e metodologias para o desenvolvimento de analisadores discursivos de nível retórico para a língua inglesa. Destacam-se os trabalhos de Marcu (1997, 2000b), Corston-Oliver (1998), Carlson e Marcu (2001), Schilder (2002), Marcu e Echihabi (2002), Soricut e Marcu (2003), Hanneforth et al. (2003), Reitter (2003) e Mahmud e Ramsay (2005).

Para possibilitar a análise discursiva automática de um texto, estes trabalhos fazem uso de uma grande variedade de conhecimentos e recursos, por exemplo, marcadores discursivos presentes no texto, informações sintáticas, aspectos da representação semântica das sentenças e dados estatísticos aprendidos automaticamente. Entre estes, os marcadores discursivos são, reconhecidamente, os maiores indicadores da estrutura discursiva de um texto e, por isso, constituem o recurso mais utilizado para possibilitar a análise automática.

Diante da importância dos marcadores discursivos, a próxima seção apresenta uma introdução a estes mecanismos lingüísticos e sua utilidade na análise discursiva. Na seção seguinte, os trabalhos anteriormente citados são revisados em ordem cronológica, com especial enfoque nos trabalhos de Marcu, por terem servido de base para este trabalho de doutorado e para vários dos outros trabalhos na área.

3.1. Marcadores Discursivos

Os marcadores discursivos são elementos coesivos formados de uma ou mais palavras que explicitam o relacionamento que existe entre as partes de um texto (Koch, 1998; Kock e Travaglia, 2002).

Os marcadores são essenciais para a análise discursiva automática, pois são os maiores indicadores superficiais das relações retóricas no texto. Por exemplo, ao se encontrar o marcador “entretanto”, “contudo” ou “porém” conectando dois segmentos textuais, há grandes chances de haver uma relação retórica de oposição (CONTRAST, ANTITHESIS ou CONCESSION) entre as proposições expressas por eles. Deve-se notar, no entanto, que não há um mapeamento unívoco entre os marcadores discursivos e as relações que sinalizam: uma mesma relação pode ser sinalizada por

vários marcadores (por exemplo, a relação CONCESSION pode ser sinalizada pelos marcadores "entretanto", "no entanto" e "mas", entre outros) e um mesmo marcador pode sinalizar várias relações (por exemplo, o marcador "porque" pode sinalizar as relações CAUSE e RESULT (volitivas ou não), JUSTIFY e EXPLANATION, entre outras).

Na Lingüística e na Lingüística Computacional, há vários trabalhos sobre marcadores discursivos e sua função no discurso. Diz-se que eles determinam a estrutura do discurso e, ao mesmo tempo, são determinados por ela. Eles são pistas que o escritor do texto deixa para que o leitor consiga, com o mínimo esforço possível, entender o relacionamento entre os significados das partes do texto e entender, portanto, o próprio sentido do texto (Koch e Travaglia, 2002; Koch, 1998). Devido a sua função, conforme os estudos conduzidos por Hirschberg e Litman (1993) e Fraser (1999), os marcadores discursivos também podem ser chamados de conectivos discursivos, operadores discursivos, partículas discursivas, sinalizadores de discurso, conectivos fáticos, conectivos pragmáticos, expressões pragmáticas, formativos pragmáticos, marcadores pragmáticos, operadores pragmáticos, partículas pragmáticas, conjuntos semânticos e conectivos de sentenças, entre outros.

Para o inglês, muitos trabalhos se destacaram no estudo dos mais diversos marcadores discursivos (por exemplo, Quirk et al., 1985; Di Eugenio, 1992, 1993; Elhadad e McKeown, 1990; Hirschberg e Litman, 1987, 1993; Knott, 1995; Knott e Dale, 1996; Knott e Mellish, 1996; Grote et. al., 1997; Fraser, 1999; Oates, 1999). Para o português do Brasil, destacam-se os trabalhos de Koch (1998), Paizan (2001) e Dias da Silva e Oliveira (2002), os quais apresentam a função de vários marcadores discursivos, seus contextos de ocorrência e que relações retóricas indicam.

Em outras teorias discursivas, a importância dos marcadores discursivos também é reconhecida. Grosz e Sidner (1986) afirmam que os marcadores discursivos também podem ser usados para indicar as relações intencionais entre as intenções subjacentes a dois segmentos. Segundo as autoras, se, entre dois segmentos, há marcadores da língua inglesa como *firstly*, *in the first place*, *second*, *then* e *lastly*, pode-se estar indicando que a intenção subjacente ao primeiro segmento textual deve ser satisfeita antes da intenção subjacente ao segundo segmento textual. Em uma argumentação diferente da tradicional, Korelsky e Kittredge (1993) afirmam que os marcadores discursivos indicam, na realidade, somente as relações semânticas entre proposições.

Um dos resultados deste trabalho de doutorado é um estudo dos marcadores discursivos e das relações retóricas que sinalizam, com base em uma análise exaustiva de um corpus de 100 textos científicos anotados retoricamente segundo a RST (Pardo e Nunes, 2004). Este corpus é descrito no próximo capítulo, juntamente com o relato do estudo conduzido.

3.2. Analisadores Discursivos Automáticos

3.2.1. Marcu (1997, 2000b): o Desenvolvimento do Primeiro Parser Retórico para o Inglês

Marcu (1997, 2000b) desenvolveu o primeiro parser retórico (conforme denominado por ele), com base na RST, para textos em inglês do gênero jornalístico, utilizando um corpus de textos anotados retoricamente chamado *RST Discourse Treebank* (Carlson et al., 2002). A metodologia que desenvolveu e a formalização que propôs formam a base da maioria dos trabalhos em análise discursiva automática.

Marcu identificou e tratou vários problemas para a automação da análise retórica, a saber:

- como delimitar os segmentos textuais que expressam proposições simples de forma consistente para que a análise retórica seja passível de automação;
- como identificar as relações retóricas intra e intersentenciais de forma automática;
- uma vez descobertas as relações retóricas, como saber que proposições são núcleos e satélites das relações;
- como construir as estruturas retóricas válidas de um texto a partir das relações retóricas entre suas proposições.

Cada uma das questões acima e as soluções propostas por Marcu são discutidas nas próximas subseções.

3.2.1.1. Delimitação das Proposições

A delimitação dos segmentos que expressam proposições simples consiste, na realidade, no conhecido problema de segmentação textual. Para uma ampla discussão sobre o assunto, veja Pardo e Nunes (2003b). Os autores apresentam uma revisão das principais técnicas da literatura para segmentação textual.

Em seus trabalhos, Marcu propôs duas soluções para este problema: uma baseada em análise de corpus e outra baseada em técnicas de Aprendizado de Máquina.

Por meio de análise de um corpus anotado retoricamente, Marcu produziu várias regras para delimitação dos segmentos em um texto. As regras se baseiam na ocorrência de sinais de pontuação no texto e de marcadores discursivos, já que estes são um dos principais indicativos superficiais da estruturação textual.

A cada padrão de itens léxicos encontrados no texto, uma ação foi associada. As ações são responsáveis por informar ao parser retórico onde inserir as marcações de início e fim de segmento. Na Tabela 3.1, mostram-se alguns exemplos de padrões lexicais e ações associadas definidos por Marcu. Na primeira linha da tabela, por exemplo, tem-se que, caso a palavra *Although* seja encontrada no início de uma sentença, o parser deve inserir uma marca de fim de segmento após a próxima vírgula que encontrar na sentença.

Tabela 3.1 – Padrões lexicais e ações para segmentação

Padrões lexicais	**Ações**
Although no começo de uma sentença	Inserir marca de fim de segmento imediatamente após a próxima ocorrência de vírgula
because no começo de uma sentença	Inserir marca de início de segmento imediatamente antes do marcador discursivo
for example no fim de uma sentença	Não inserir marca de segmento alguma

Com esta técnica, Marcu atingiu precisão (isto é, a medida tradicional *precision*) de 90% e cobertura (isto é, a medida tradicional *recall*) de 81%. Neste contexto, precisão indica quantos segmentos corretos foram detectados em relação a tudo que foi detectado e cobertura indica quantos segmentos corretos foram detectados em relação a tudo que deveria ter sido detectado.

Na outra abordagem, utilizando técnicas de Aprendizado de Máquina, mais especificamente, o classificador C4.5 (Quinlan, 1993), Marcu lista as *features* (características) abaixo como sendo as principais para determinar se um item lexical do texto indica ou não a presença de uma marca de segmento:

- a classe gramatical do item lexical sob análise;
- as classes gramaticais dos dois itens lexicais que precedem e seguem o item lexical sob análise;
- se o item lexical sob análise é (parte de) um marcador discursivo;
- se o item lexical é uma abreviatura;
- se há verbos nas proximidades do item lexical sob análise.

Com esta técnica, Marcu conseguiu um desempenho (*F-measure* – uma combinação das medidas de precisão e cobertura – que é uma medida única do quão próximo do ideal um sistema está) de 97%.

3.2.1.2. Determinação das Relações Retóricas

Para determinar as relações retóricas entre as proposições expressas em um texto, Marcu faz uso dos marcadores discursivos presentes no texto.

Para identificar os marcadores discursivos e diferenciá-los de marcadores sentenciais e pragmáticos, Marcu utiliza padrões lexicais, também obtidos por meio de análise de corpus, semelhantes aos padrões mostrados na Tabela 3.1. Marcadores sentenciais e pragmáticos possuem formação semelhante aos marcadores discursivos, mas se distinguem pelo fato de não refletirem a estrutura discursiva do texto. Os marcadores sentenciais são utilizados em uma sentença para conectar suas partes somente, sem função discursiva. Por exemplo, o "e" que forma o sujeito composto da sentença "João e Maria são irmãos." é um marcador sentencial. Marcadores pragmáticos, por sua vez, remetem o leitor a seu conhecimento de mundo. Por exemplo, na sentença "João foi preso de novo.", o marcador "de novo" leva o leitor a inferir que João já foi preso antes.

Pela análise de corpus que realizou, Marcu associou a cada marcador discursivo as possíveis relações retóricas sinalizadas. Por exemplo, o marcador *although*, dependendo da posição em que ocorre na sentença, pode indicar a relação

retórica CONCESSION ou CONTRAST entre as proposições expressas pelos segmentos em que o marcador é observado. Com isso, durante a análise automática de um texto, todas as relações retóricas possíveis entre proposições expressas por segmentos com marcadores discursivos são listadas.

Nos casos em que não há marcadores discursivos entre segmentos textuais, Marcu tenta inferir a relação retórica pela aplicação de algumas heurísticas simples. Por exemplo: se um segmento repete algumas palavras do segmento anterior e não há marcadores discursivos entre eles, então se estabelece uma relação BACKGROUND; caso contrário, estabelece-se uma relação ELABORATION, que é a relação mais comum e genérica no elenco de relações da RST. Com esta técnica, Marcu conseguiu precisão de 78% e cobertura de 47%. Neste caso, a precisão indica quantas relações retóricas foram corretamente identificadas em relação a tudo que se identificou e cobertura indica quantas relações retóricas foram corretamente identificadas em relação a tudo que deveria ter sido identificado.

3.2.1.3. Determinação dos Núcleos e Satélites

Uma vez que os segmentos textuais que expressam proposições simples e as relações retóricas entre elas são identificados, Marcu utiliza a ordem preferencial de realização de núcleos e satélites das relações retóricas para determinar que proposições são núcleos e que proposições são satélites.

Para determinar a ordem entre o núcleo e o satélite de cada relação, Marcu recorreu a sua análise de corpus e associou aos marcadores discursivos estudados as possíveis ordenações entre os segmentos. Por exemplo, para o marcador *Although* que ocorre no começo de uma sentença, a proposição expressa pelo segmento ao qual o marcador pertence é classificada como satélite e a proposição expressa pelo segmento seguinte como núcleo.

Esta técnica atingiu precisão de 85% e cobertura de 50%. Neste caso, precisão indica quantos núcleos e satélites foram identificados corretamente em relação a tudo que foi identificado e cobertura indica quantos núcleos e satélites foram identificados corretamente em relação a tudo que deveria ter sido identificado.

3.2.1.4. Construção das Estruturas Retóricas Válidas

Por fim, Marcu abordou o problema que considerou um dos mais desafiadores: construir as estruturas retóricas válidas de um texto a partir das relações retóricas que se estabelecem entre suas proposições. Estruturas inválidas são aquelas em que as relações retóricas utilizadas não são adequadas para as proposições/subestruturas sendo relacionadas.

Segundo Marcu, a falta de formalização da RST não permitia que se automatizasse este passo. Marcu procedeu, então, a uma completa formalização da RST. Desta formalização, os pontos principais e inovadores que permitiram a automação da análise retórica são:

1. critério da composicionalidade: dadas duas estruturas retóricas RSTree1 e RSTree2:

   RSTree1 = rhet_rel(R1,S1,S2)
   RSTree2 = rhet_rel(R2,S3,S4)

   em que o predicado rhet_rel(R,Y,X) representa uma estrutura retórica na qual a proposição (ou subestrutura) Y é o satélite e a proposição (ou subestrutura) X o núcleo na relação retórica R, é possível combinar RSTree1 e RSTree2 em uma estrutura maior RSTree3 por meio da relação retórica R3, se R3 se estabelece entre os núcleos das estruturas a serem combinadas, ou seja, se R3 se estabelece entre as proposições nucleares (ou entre os núcleos das subestruturas) S2 da RSTree1 e S4 da RSTree2.

   Esta definição, quando necessária, é aplicada recursivamente até que se chegue aos nós terminais das estruturas a serem combinadas. Como exemplo, considere o texto mostrado na Figura 3.1, o qual foi retirado integralmente do trabalho de Marcu (2000b), segmentado e numerado para referência.

[*No matter how much one wants to stay a nonsmoker,*]$_1$ [*the truth is that the pressure to smoke in junior high is greater than it will be any other time of one's life.*]$_2$ [*We know that 3.000 teens start smoking each day,*]$_3$ [*although it is a fact that 90% of them once thought that smoking was something that they'd never do.*]$_4$

Figura 3.1 – Texto-exemplo de Marcu (2000b)

Supondo que as seguintes relações retóricas, correspondentes às relações retóricas elementares que conectam diretamente duas proposições, tenham sido reconhecidas:

rhet_rel(JUSTIFY,1,2)
rhet_rel(CONCESSION,4,3)
rhet_rel(EVIDENCE,3,2)
rhet_rel(RESTATEMENT,4,1)

Uma possível estrutura retórica para o texto é a mostrada na Figura 3.2: tem-se a subestrutura conectando as proposições 1 e 2 pela relação JUSTIFY; tem-se a subestrutura conectando as proposições 4 e 3 pela relação CONCESSION; para montar uma estrutura maior que abranja as subestruturas anteriores, é necessário encontrar uma relação retórica que conecte os núcleos das duas, que, neste caso, é a relação EVIDENCE, a qual conecta as proposições 2 (núcleo da primeira subestrutura) e 3 (núcleo da segunda subestrutura). Como a relação RESTATEMENT estabelece-se entre proposições que não são os núcleos das subestruturas, ela não pode ser usada no lugar de EVIDENCE para montar a estrutura final (caso contrário, uma estrutura inválida seria produzida).

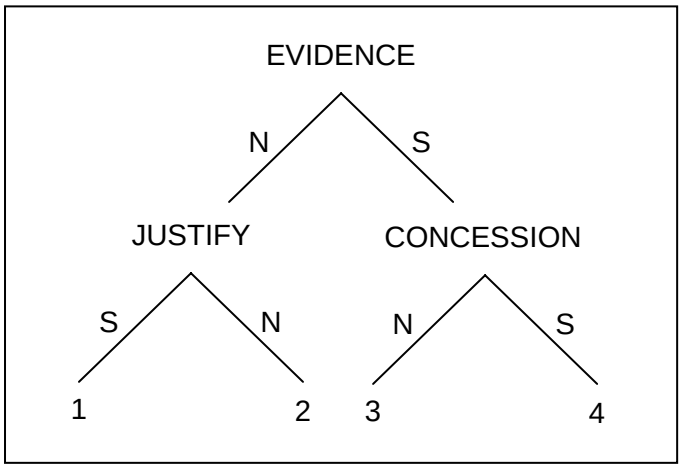


Figura 3.2 – Possível estrutura retórica para texto da Figura 3.1

2. com base no critério da composicionalidade, dado o conjunto de relações retóricas que se estabelecem entre as proposições (como no exemplo anterior), é possível construir várias estruturas retóricas válidas para um mesmo texto. Para que fosse possível construir "todas" as estruturas retóricas válidas, e somente as válidas, Marcu desenvolve vários algoritmos. O mais eficiente deles, utilizando uma gramática, produz todas as estruturas com tempo de execução linear em relação ao número de segmentos identificados no texto. Neste algoritmo, dado o conjunto de relações retóricas entre os segmentos, é produzida uma gramática na forma normal de Chomsky que produz todas as combinações possíveis entre as proposições (ou subestruturas). Ao se executar esta gramática, todas as estruturas retóricas válidas são construídas. Na Figura 3.3, reproduz-se o algoritmo proposto por Marcu.

Input: a sequence U=1,2,...,N of elementary textual units and a set RR of rhetorical relations that hold among these units.
Output: a grammar in Chomsky normal form that can be used to derive all and only the parse trees that correspond to the valid text structures of U.

1. **for** i:=1 **to** N
2. add rules S→i, S(i,i,nucleus,leaf,{i})→i, S(i,i,satellite,leaf,{i})→i
3. **endfor**
4. **for** size_of_span:=1 **to** N-1
5. **for** l:=1 **to** N-size_of_span
6. h:=l+size_of_span
7. **for** b:=l **to** h-1
8. **for** x:=l **to** b
9. **for** y:=b+1 **to** h
10. **for each** $name_1$ for which a rule has S(l,b.satellite,$name_1$,{x}) as its head
11. **for each** $name_2$ for which a rule has S(b+1,h,nucleus,$name_2$,{y}) as its head
12. **for each** mononuclear relation name such that rhet_rel(name,x,y)∈RR or rhet_rel(name,[l,b],[b+1,h])∈RR
13. add rule S→S(l,b,satellite,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
14. add rule S(l,h,satellite,name,{y})→S(l,b,satellite,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
15. add rule S(l,h,nucleus,name,{y})→S(l,b,satellite,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
16. **endfor**
17. **endfor**
18. **endfor**
19. **for each** $name_1$ for which a rule has S(l,b.nucleus,$name_1$,{x}) as its head
20. **for each** $name_2$ for which a rule has S(b+1,h,satellite,$name_2$,{y}) as its head
21. **for each** mononuclear relation name such that rhet_rel(name,y,x)∈RR or rhet_rel(name,[b+1,h],[l,b])∈RR
22. add rule S→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,satellite,$name_2$,{y})
23. add rule S(l,h,satellite,name,{x})→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,satellite,$name_2$,{y})
24. add rule S(l,h,nucleus,name,{x})→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,satellite,$name_2$,{y})
25. **endfor**
26. **endfor**
27. **endfor**
28. **for each** $name_1$ for which a rule has S(l,b.nucleus,$name_1$,{x}) as its head
29. **for each** $name_2$ for which a rule has S(b+1,h,nucleus,$name_2$,{y}) as its head
30. **for each** multinuclear relation name such that rhet_rel(name,x,y)∈RR or rhet_rel(name,[l,b],[b+1,h])∈RR
31. add rule S→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
32. add rule S(l,h,satellite,name,{x,y})→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
33. add rule S(l,h,nucleus,name,{x,y})→S(l,b,nucleus,$name_1$,{x}) S(b+1,h,nucleus,$name_2$,{y})
34. **end all for loops**

Figura 3.3 – Algoritmo de Marcu para construção de estruturas retóricas válidas

Neste algoritmo, pelos passos 1-3, produzem-se regras que levam aos nós terminais da estrutura retórica, isto é, as proposições; pelos passos 10-18,

produzem-se regras que geram o satélite antes do núcleo para uma determinada relação retórica; pelos passos 19-27, produzem-se regras que geram o núcleo antes do satélite para uma determinada relação retórica; por fim, a partir do passo 28, produzem-se regras que geram relações multinucleares.

Como exemplo, considere uma relação CONCESSION entre proposições expressas pelos segmentos 1 e 2, com o segmento 2 sendo o núcleo da relação. Basicamente, as regras produzidas por esse algoritmo para produzir tal análise seriam:

```
S(1,2,satellite,CONCESSION,{2}) -->
                S(1,1,satellite,leaf,{1}),
                S(2,2,nucleus,leaf,{2}).
S(1,2,nucleus,CONCESSION,{2}) -->
                S(1,1,satellite,leaf,{1}),
                S(2,2,nucleus,leaf,{2}).
S(1,1,satellite,leaf,{1}) --> 1.
S(2,2,nucleus,leaf,{2}) --> 2.
```

Os argumentos do termo S à esquerda de cada regra de produção especificam, nesta ordem: o primeiro segmento abrangido pela subestrutura retórica sendo construída neste ponto; o último segmento abrangido pela subestrutura retórica sendo construída neste ponto; a nuclearidade do segmento, isto é, se ele é núcleo (*nucleus*) ou satélite (*satellite*) da relação; a relação que se estabelece entre os segmentos (o termo *leaf* – "folha" – especifica que o segmento é um nó terminal da estrutura, ou seja, uma proposição), e o conjunto de segmentos mais nucleares da subestrutura em questão. O conjunto de segmentos mais nucleares é necessário para que se possa verificar o critério da composicionalidade, isto é, se uma relação estabelece-se entre os núcleos das subestruturas que relaciona, quando este é o caso. Os termos à direita das regras de produção especificam como a estrutura em questão é composta, isto é, quais outros segmentos e/ou subestruturas a compõem. É interessante notar que as duas primeiras regras especificadas acima se diferenciam pela nuclearidade atribuída à subestrutura em foco, que pode ser núcleo ou satélite de uma estrutura maior, caso uma estrutura maior exista.

Para o conjunto de relações retóricas do texto da Figura 3.1, pela aplicação do algoritmo de Marcu, a estrutura retórica da Figura 3.4 também poderia ser produzida e, como se pode verificar, pelo critério da composicionalidade, é uma estrutura válida.

As duas estruturas são, de fato, todas as estruturas válidas possíveis de serem construídas com as relações utilizadas.

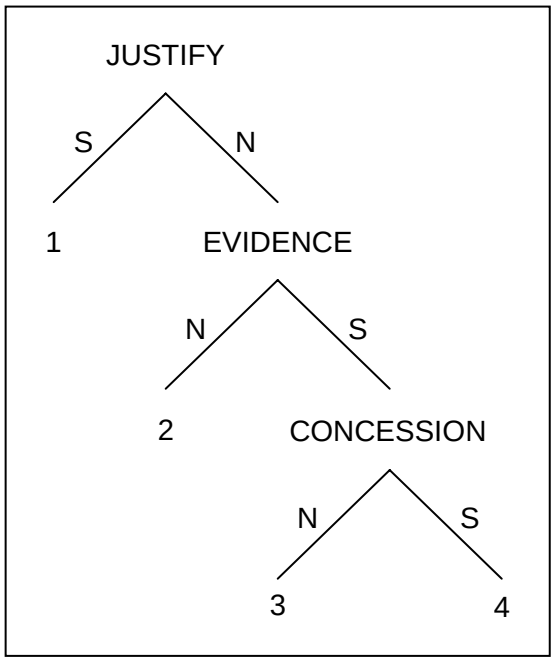


Figura 3.4 – Outra possível estrutura retórica para o texto da Figura 3.1

3.2.2. Corston-Oliver (1998): o Analisador RASTA

O analisador retórico produzido por Corston-Oliver, chamado RASTA (*Rhetorical Structure Theory Analyzer*), foi desenvolvido a partir da análise de textos enciclopédicos. Ele se baseou na formalização proposta por Marcu, discutida anteriormente.

Este trabalho merece destaque por abordar, além dos marcadores discursivos, aspectos das formas lógicas e informações das estruturas sintáticas lexicalizadas das sentenças de um texto para determinar as relações retóricas entre suas proposições. Dentre as informações disponíveis, Corston-Oliver verifica as seguintes informações para a análise automática, principalmente:

- se as proposições a serem relacionadas têm seus segmentos correspondentes subordinados sintaticamente uns aos outros;
- se as proposições têm seus segmentos correspondentes na voz ativa ou passiva;
- presença de fenômenos lingüísticos como anáforas e elipses nos segmentos que expressam as proposições;
- se os núcleos dos constituintes sintáticos dos segmentos que expressam as proposições são os mesmos ou não;
- as classes gramaticais das palavras dos segmentos que expressam as proposições.

Como exemplo, na Figura 3.5, mostram-se os critérios necessários listados por Corston-Oliver para que se estabeleça a relação retórica CAUSE entre duas proposições expressas por dois segmentos quaisquer A e B de um texto (em inglês).

1. O segmento A precede o segmento B no texto
2. O segmento A não é sintaticamente subordinado ao segmento B
3. O segmento B não é sintaticamente subordinado ao segmento A
4. O sujeito do segmento B é um pronome demonstrativo ou é modificado por um demonstrativo; ou os segmentos A e B são coordenados por um símbolo de dois pontos (;)
5. O segmento B está na voz passiva e possui a palavra *cause*; ou o segmento B contém a frase indicativa *result from* com o verbo estando possivelmente flexionado

Figura 3.5 – Critérios de Corston-Oliver para a relação retórica CAUSE

É importante dizer que Corston-Oliver também trabalhou com o conceito de subespecificação, isto é, quando, por falta de informação disponível, não se consegue determinar a relação retórica que se estabelece entre duas proposições quaisquer, e, por esta razão, indica-se na estrutura retórica que a relação existe, mas não se especifica qual relação há entre as proposições.

Corston-Oliver não apresenta a avaliação do RASTA.

3.2.3. Carlson e Marcu (2001): um Manual para Segmentação Textual

Após coletar e revisar todo o conhecimento sobre segmentação textual gerado pelas técnicas de análise discursiva de Marcu (1997, 2000b), Carlson e Marcu produziram

um manual para segmentação de textos em inglês de forma que se delimite as proposições simples expressas. Por este manual, é possível determinar exatamente e, mais importante, de forma consistente, como reconhecer automaticamente segmentos textuais que expressam proposições simples.

Segundo as regras propostas pelos autores, em um texto, devem ser identificados como segmentos:

1. orações principais;
2. orações subordinadas com marcadores discursivos;
3. complementos oracionais de verbos atributivos, isto é, verbos que atribuem uma expressão (como falas ou pensamentos) a algo ou alguém;
4. orações coordenadas;
5. orações temporais, isto é, orações que expressam o tempo/momento em que um evento ocorreu;
6. orações relativas.

Ainda, segundo os autores, não devem ser identificados como segmentos:

1. orações que cumprem as funções de sujeito ou objeto dos verbos;
2. complementos verbais oracionais.

3.2.4. Schilder (2002): o Uso de Técnicas de Recuperação de Informação

O trabalho de Schilder destaca-se pelo uso que faz de técnicas de Recuperação de Informação (RI) para auxiliar a análise retórica. Em uma primeira etapa de seu sistema de análise retórica, dado um texto, sua estrutura retórica parcial é produzida por meio da identificação dos marcadores discursivos presentes e das relações que estes indicam. A estrutura parcial é então completada pelo uso de técnicas de RI.

Uma estrutura parcial não contém todos os segmentos delimitados em um texto. Isso ocorre pela falta de informação disponível (por exemplo, ausência de marcadores discursivos e ambigüidade retórica) para determinar onde os segmentos seriam anexados na estrutura retórica. Para decidir onde anexar os segmentos restantes na estrutura, Schilder utiliza informação sobre a topicalidade dos segmentos. Segmentos topicais são mais importantes para um texto e, portanto, devem ser

anexados em posições mais importantes da estrutura retórica, ou seja, em posições mais nucleares.

Para determinar a topicalidade dos segmentos restantes de um texto, Schilder os representa em vetores, seguindo a proposta do modelo vetorial de Salton (1971), e os compara com o vetor do título do texto sob análise. Os segmentos cujos vetores são mais próximos do vetor do título do texto recebem uma maior pontuação e são considerados topicais, restringindo, assim, os locais aos quais estes segmentos podem ser anexados à estrutura retórica parcial.

É importante ressaltar que, como Corston-Oliver, Schilder também permite subespecificação na estrutura retórica que produz para um texto. Desta forma, apesar de conseguir determinar a correta localização dos segmentos textuais na estrutura retórica, as relações podem não ser especificadas.

Schilder não avaliou o desempenho de seu analisador discursivo, mas o validou em uma aplicação de sumarização automática de textos, conseguindo bons resultados.

3.2.5. Marcu e Echihabi (2002): uma Abordagem *Bayesiana* para o Reconhecimento de Relações Discursivas

Marcu e Echihabi utilizam um classificador *naive-bayes* (Mitchell, 1997) para determinar as relações retóricas entre duas proposições para textos do gênero jornalístico. Como *features* para o aprendizado, eles utilizaram as próprias palavras dos segmentos que expressam as proposições envolvidas no processo. Com isso, eles visam a capturar dois tipos de conhecimento: (i) que relações retóricas são indicadas pelos marcadores discursivos e (ii) conhecimento de mundo. Em relação ao conhecimento de mundo, considere o exemplo abaixo dado pelos autores, no qual há uma relação retórica CONTRAST entre as proposições expressas pelas sentenças:

John is good in Math and Science. Paul fails almost every class he takes.

Por este exemplo, o classificador *bayesiano* aprenderia a relação de oposição que há entre as palavras *good* e *fail*. Ao se deparar com um novo exemplo que contivesse estas palavras, o classificador conseguiria, então, inferir a relação CONTRAST.

Apesar de ser uma abordagem promissora, Marcu e Echihabi a utilizam para determinar um pequeno conjunto de 4 relações bem distintas (CONTRAST, EXPLANATION, CONDITION e ELABORATION), atingindo um desempenho de 49%. Para o conjunto completo de relações, levando-se em consideração que algumas relações possuem diferenças de definição muito tênues, os autores supõem que essa abordagem não seria informada o suficiente para diferenciar as relações com grande precisão.

3.2.6. Soricut e Marcu (2003): Modelos Probabilísticos com Base em Informação Sintática e Lexical

Soricut e Marcu utilizam modelos probabilísticos baseados em informações sintáticas e lexicais para realizar a análise discursiva automática intra-sentencial para textos do gênero jornalístico. Os autores propõem modelos distintos para realizar a segmentação textual e a detecção de relações retóricas.

O modelo probabilístico para segmentação textual é treinado com as palavras consideradas núcleos (*heads*) dos constituintes (sujeito, objetos, predicativos, etc.) das estruturas sintáticas lexicalizadas das sentenças. Depois de treinado, dada uma nova sentença, o modelo a segmenta nos pontos mais prováveis. Por exemplo, dada uma sentença S formada pelas palavras w_1, w_2 ... w_n, a probabilidade da palavra w_i (com $1 \leq i \leq n$) indicar o início de um novo segmento é obtida pela combinação da probabilidade (a) da própria palavra w_i, (b) da palavra que é núcleo do constituinte sintático a que w_i pertence e (c) da palavra que é núcleo do constituinte sintático que domina o constituinte sintático a que w_i pertence indicarem um novo segmento. Com esta técnica, Marcu atingiu um desempenho de 84%.

Para determinação das relações retóricas, os autores treinam um modelo probabilístico com os núcleos (*heads*) dos segmentos contidos pelas sentenças (na maioria dos casos, os verbos) classificados com as relações retóricas que se estabelecem entre eles, para, então, determinar a estrutura retórica entre segmentos de novas sentenças com base em seus núcleos. Os autores conseguiram um desempenho médio de 75% com esta técnica.

3.2.7. Reitter (2003): *Support Vector Machines* para Análise Discursiva

Reiter utilliza a técnica de Aprendizado de Máquina *Support Vector Machine* (SVM) (Vapnik, 1995) e desenvolve classificadores para a realização da análise discursiva automática.

Nesse trabalho, as estruturas retóricas são representadas segundo o esquema *Underspecified Rhetorical Markup Language* (URML), desenvolvido por Reitter e Stede (2003) com base em XML (*eXtensible Markup Language*). Segundo seus autores, esse esquema permite a representação de relações discursivas de qualquer teoria, pois é genérico e adaptável. Como exemplo, mostra-se, abaixo, como se pode representar a relação retórica CONCESSION, do tipo mononuclear, entre os segmentos A e B, com B sendo o núcleo da relação, produzindo uma estrutura retórica chamada C.

```
<mononuclearRelation type="CONCESSION" id="C">
        <satellite id="A" />
        <nucleus id="B" />
</mononuclearRelation>
```

Para detectar as relações retóricas, Reitter utiliza uma combinação de SVMs, com cada uma das SVMs treinada para identificar uma relação retórica específica. Desta forma, diante de dois segmentos, todas as SVMs são aplicadas e a probabilidade da ocorrência de cada uma das relações retóricas existentes são produzidas. A relação de maior probabilidade é escolhida.

Para dois segmentos quaisquer, para os quais se deseja identificar a relação retórica, as *features* utilizadas para o aprendizado e classificação são:

- presença de marcadores discursivos e pronomes nos segmentos;
- presença de descrições definidas (um tipo de anáfora) nos segmentos;
- sinais de pontuação dos segmentos;
- etiquetas morfossintáticas das palavras na fronteira entre os segmentos;
- similaridade lexical entre os segmentos, calculada por meio da verificação das palavras em comum entre os segmentos (conforme feito por Hearst, 1997);
- tamanho dos segmentos.

De acordo com o autor, essas *features* codificam conhecimento lingüístico relevante para a tarefa em questão e se mostraram importantes durante a classificação.

Na avaliação conduzida para textos jornalísticos em inglês (os mesmos utilizados por Marcu em sua avaliação), Reitter relata um desempenho médio de 61,8%. Para um corpus em alemão, também composto por textos jornalísticos, o desempenho médio é de 39,1%. O autor atribui essa diferença de desempenho entre as línguas à quantidade menor de dados de treinamento para a língua alemã.

3.2.8. Hanneforth et al. (2003): uma Gramática para Análise Discursiva

Hanneforth et al. propõem uma gramática para a realização da análise retórica. Na gramática especificada na linguagem de programação Prolog, produzem-se regras que reconhecem os segmentos textuais e as relações existentes entre suas proposições através da verificação da presença de marcadores discursivos e sinais de pontuação. O uso de uma gramática de nível textual é o principal diferencial deste trabalho em relação aos outros. Considere a regra gramatical abaixo como exemplo:

```
rst({cat: main_clause, rel: CONCESSION, discourse_particle: no, type: nuc_sat}) -->
        rst({cat: subordinate_clause, discourse_particle: even_though, role: satellite}),
        rst({cat: main_clause, discourse_particle: no, role: nucleus}).
```

Basicamente, segundo essa regra, haverá uma relação CONCESSION entre dois segmentos, com o primeiro sendo o satélite e o segundo o núcleo da relação, se (a) o primeiro for uma oração subordinada ao segundo e contenha o marcador discursivo *even though* e (b) o segundo segmento for a oração principal e não tiver um marcador discursivo.

Os autores permitem que, nas estruturas retóricas produzidas, haja ambigüidade, representada pela presença de várias relações entre duas ou mais proposições.

A representação de estruturas retóricas é feita em URML, esquema desenvolvido por Reitter e Stede (2003), apresentado na subseção anterior. Em casos

em que não há marcadores discursivos para que se determinem as relações retóricas, aplica-se uma regra padrão da gramática que especifica a relação retórica ELABORATION ou SEQUENCE entre as proposições envolvidas.

Por ser uma proposta de abordagem ao problema da análise discursiva automática, os autores não relatam a avaliação do uso da gramática.

3.2.9. Mahmud e Ramsay (2005): Análise Discursiva para Textos de Qualidade Duvidosa

Mahmud e Ramsay desenvolveram um módulo de análise discursiva automática para um sistema de auxílio ao aprendizado de redação por alunos de nível médio para a língua inglesa.

O sistema desenvolvido pelos autores indica possíveis relações retóricas entre as proposições expressas pelas sentenças de uma redação feita por um aluno. Devido a isto, não se pode assumir que os textos a serem analisados pelo analisador discursivo são bons. Esses textos podem conter erros gramaticais, passagens incoerentes e organização discursiva inadequada.

Para lidar com esses problemas, os autores fizeram uso de outros recursos para a análise discursiva além dos marcadores discursivos presentes nos textos, que, segundo eles, são escassos. Utilizando o ambiente WEKA (Witten e Frank, 2000), que é um sistema que contém diversos algoritmos de Aprendizado de Máquina, os autores testaram vários algoritmos com as seguintes *features*:

- presença de marcadores discursivos nas sentenças;
- presença de termos anafóricos nas sentenças;
- modalidade das sentenças (indicada pelos verbos modais da língua inglesa);
- verbo principal das sentenças;
- distância entre as sentenças a serem relacionadas;
- relação entre os verbos principais das sentenças a serem relacionadas, utilizando-se a relação obtida na WordNet[2] (Fellbaum, 1998);
- a conexão referencial entre as duas sentenças, conforme proposto por Grosz et al. (1995).

[2] http://wordnet.princeton.edu/

Buscou-se identificar um pequeno grupo de relações bastante distintas, a saber: SEQUENCE, ELABORATION, CONTRAST, "outra relação" ou nenhuma relação. Permitir que nenhuma relação seja detectada é necessário pelo fato de a qualidade dos textos a serem analisados ser duvidosa.

Ao ser treinado e testado com um corpus de redações anotadas discursivamente por um único especialista em análise discursiva, o analisador discursivo proposto atingiu um desempenho máximo de 88,4%. Os autores verificaram que as *features* mais relevantes para a classificação são a presença de termos anafóricos nas sentenças e a distância entre as sentenças relacionadas.

Como será relatado no próximo capítulo, neste trabalho de doutorado, segue-se, principalmente, a abordagem de Marcu (1997, 2000b) para a realização da análise discursiva pelo DiZer. Utiliza-se, também, o manual para segmentação textual desenvolvido por Carlson e Marcu (2001).

No próximo capítulo, apresentam-se o DiZer, seu processo de desenvolvimento e sua avaliação.

4. DiZer: Um Analisador Discursivo Automático para o Português do Brasil

O analisador discursivo automático DiZer (*DIscourse analyZER*) para o português do Brasil desenvolvido neste trabalho de doutorado faz parte da abordagem conhecida como simbólica na área de Inteligência Artificial. Nesta abordagem, explicita-se e formaliza-se o conhecimento necessário para que se automatize a tarefa em foco, de forma que esse conhecimento seja passível de leitura e interpretação por um humano.

Para o desenvolvimento do DiZer, o conhecimento visado é o conhecimento lingüístico que possibilite a análise retórica automática, em particular, os marcadores textuais indicadores da estrutura retórica dos textos, mais especificamente, marcadores discursivos e palavras e frases indicativas. Como já discutido no capítulo anterior, na Seção 3.1, os marcadores discursivos são claros sinalizadores da estrutura discursiva dos textos. Palavras e frases indicativas, por sua vez, são, segundo Paice (1981), conjuntos de palavras que indicam o conteúdo do segmento textual em que ocorrem. Por exemplo, em textos científicos, geralmente, a frase indicativa "O objetivo deste trabalho é" no início de uma sentença indica ao leitor do texto que a sentença contém o objetivo do trabalho sendo relatado e que, provavelmente, há uma relação retórica PURPOSE entre a proposição expressa por esta sentença e alguma proposição expressa por um segmento adjacente. De forma similar, a ocorrência de palavras como "vantagens", "positivo" e "negativo" em um segmento indicam que este apresenta a avaliação de algo e, portanto, pode sinalizar uma relação EVALUATION entre a proposição expressa por este segmento e alguma proposição expressa por um segmento adjacente.

Para a compilação do conhecimento necessário, foi analisado um corpus de 100 **T**extos **C**ientíficos da **C**omputação anotados retoricamente segundo a RST, chamado CorpusTCC. Estes textos foram coletados do Corpus NILC (Pinheiro e Aluísio, 2003) e do CorpusDT (Feltrim et al., 2001). Optou-se pela análise de textos do gênero científico pelo fato de estes possuírem marcadores textuais em número significativo, serem supostamente bem escritos e outros trabalhos sobre discurso para o português utilizarem textos desse gênero (por exemplo, Rino, 1996; Feltrim et al., 2001; Pardo, 2002). Devido a isso, diz-se que o DiZer é, primariamente, um analisador discursivo para textos científicos. Optou-se por textos do domínio da Computação devido à disponibilidade destes textos e à familiaridade do autor com o

domínio, o que possibilita a realização de uma análise mais informada e consistente. Apesar de tais escolhas, como será discutido na Seção 4.3, o DiZer também pode ser aplicado a textos de outros gêneros e domínios, dado que, em geral, grande parte dos marcadores textuais são independentes de gênero e domínio.

Com base na análise conduzida, produziu-se um repositório de informação discursiva que consiste na principal componente do DiZer. Este repositório contém cerca de 740 padrões de análise que especificam o relacionamento entre relações retóricas e seus marcadores textuais. Com o uso destes padrões, por meio de um processo de casamento de padrões, isto é, o processo pelo qual procura-se por instâncias compatíveis com algum padrão (Russell e Norvig, 2003), pode-se realizar a análise retórica automática de textos. Esse processo é o principal processo do DiZer e é descrito na Seção 4.2, na qual descrevem-se o DiZer, seus principais módulos e repositórios de informação. Na próxima seção, relatam-se as etapas de construção, anotação e extração de conhecimento do CorpusTCC. A avaliação do DiZer é apresentada na Seção 4.3.

4.1. Análise de Corpus

4.1.1. Descrição do CorpusTCC

Para compor o CorpusTCC, foram coletadas do Corpus NILC (Pinheiro e Aluísio, 2003) e do CorpusDT (Feltrim et. al., 2001) 47 introduções de dissertações de Mestrado e 3 introduções de qualificações de Mestrado, ambos os casos do domínio da Ciência da Computação, resultando, no total, em 50 introduções, em formato *plain text*, com tamanho de 1 a 4 páginas. Esse corpus contém, aproximadamente, 53.000 palavras e 1.350 sentenças.

Coletaram-se somente as introduções dos textos, em vez dos textos completos, pelas seguintes razões: (a) as introduções são suficientes para cumprir a finalidade a qual o corpus serve; (b) anotar retoricamente dissertações e qualificações completas é inviável devido à complexidade desta tarefa e o tempo que consome.

As introduções são de várias áreas da Computação, a saber: banco de dados (bd), engenharia de software (es), hipermídia (h), inteligência computacional (ic) e sistemas distribuídos (sd). Para compor o corpus, essas introduções foram

subdivididas de acordo com suas seções, também visando a simplificar a anotação retórica. Com isso, o corpus contém, no total, 100 textos, sendo que cada texto tem, ao final, tamanho máximo de 3 páginas, variando de 1 a 12Kb em tamanho de arquivo (totalizando 343Kb). Na Tabela 4.1, mostra-se o número de textos e de palavras por área, indicando, também, a número médio de palavras por texto. Na Figura 4.1, mostra-se a distribuição dos textos por área.

Tabela 4.1 – Número de textos por área

Área	Número de textos	Número de palavras	Número médio de palavras por texto
bd	7	2.859	408
es	33	16.084	487
h	29	16.980	585
ic	20	9.549	477
sd	11	7.172	652

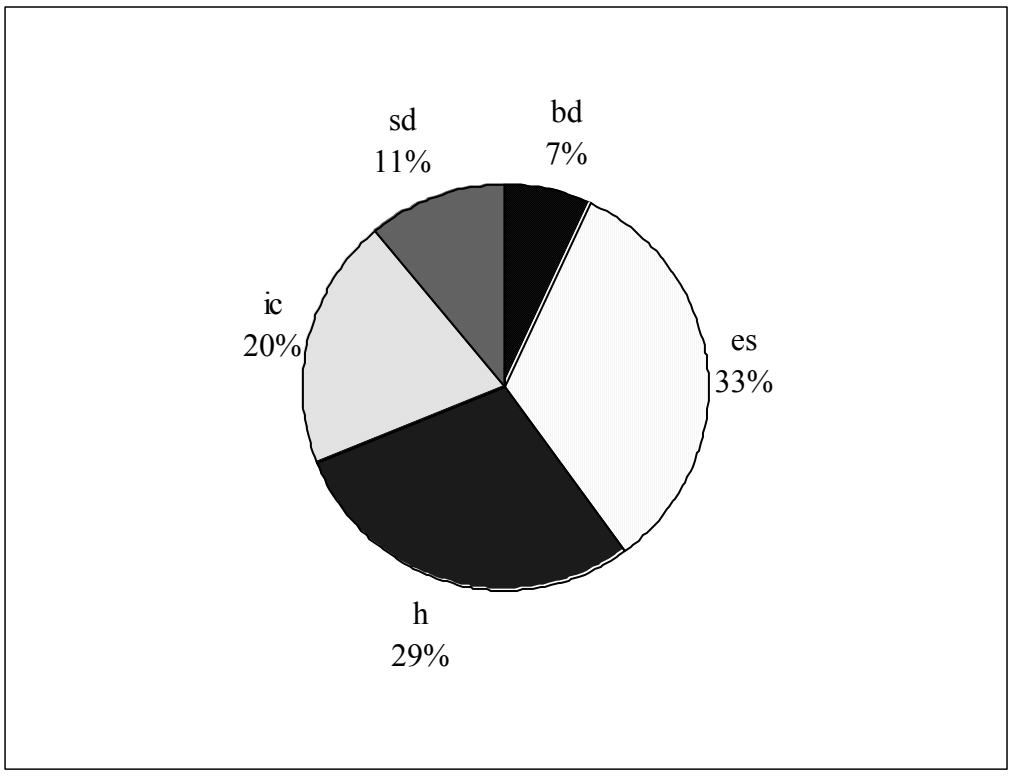


Figura 4.1 – Distribuição dos textos por área no CorpusTCC

O padrão utilizado para a nomeação dos 100 arquivos contendo os textos do corpus é o seguinte:

"Dissertacao_/Qualificacao_" + ÁREA + ID_TEXTO + "_parte" + ID_SEÇÃO

em que ÁREA é indicada pelas siglas das áreas da computação (bd, es, h, ic ou sd), ID_TEXTO é um identificador único para o texto da área em questão e ID_SEÇÃO indica à qual seção da introdução o texto corresponde. Por exemplo, uma introdução de uma dissertação da área de banco de dados que contenha 2 seções originaria dois arquivos chamados "Dissertacao_bd1_parte1.txt" e "Dissertacao_bd1_parte2.txt".

4.1.2. Anotação Retórica do CorpusTCC

O CorpusTCC foi anotado retoricamente por somente um anotador especialista em RST (o autor desta tese) para manter a consistência da anotação. Pelos mesmos motivos, essa estratégia de anotação é recorrente na literatura (veja, por exemplo, Williams e Reiter, 2003).

4.1.2.1. Ferramenta de Anotação do CorpusTCC

Para anotar os textos do CorpusTCC, foi utilizada a ferramenta de edição gráfica *RST Annotation Tool* de Marcu , que é uma variação da ferramenta RSTTool de O'Donnel (1997). Por meio dessa ferramenta, é possível efetuar todos os passos da anotação retórica, ou seja, basicamente, segmentar o texto, escolher as relações retóricas e determinar o que é núcleo e satélite. A ferramenta também oferece possibilidade de escolher o elenco de relações retóricas a ser utilizado, desfazer operações e modificar estruturas retóricas já prontas, entre outras opções. Por fim, mas não menos importante, utilizando-se a ferramenta, é possível anotar um texto retoricamente por meio de diversas estratégias, conforme discutido por Carlson e Marcu (2001). A estratégia de marcação adotada para o CorpusTCC é discutida posteriormente.

Conforme o texto é segmentado e anotado retoricamente, a *RST Annotation Tool* armazena automaticamente os passos executados pelo usuário, permitindo que se analise o histórico de anotação de um texto, caso seja necessário. Além disso, diferentemente da ferramenta de O'Donnel, na ferramenta modificada de Marcu, ao final da anotação retórica, é oferecida ao usuário a possibilidade de armazenar a análise feita em arquivos de texto nos formatos SGML ou da linguagem de programação LISP.

A possibilidade de trabalhar com os textos visualmente, explorando toda a facilidade da manipulação gráfica, e armazenar as análises feitas em arquivos de texto, o que facilita a manipulação computacional posterior, constituíram a grande motivação para a marcação do CorpusTCC pela *RST Annotation Tool*.

4.1.2.2. Segmentação Textual

Para a anotação do CorpusTCC, seguindo o que se tem feito na maioria dos trabalhos recentes de análise retórica automática (veja capítulo anterior), adotou-se, basicamente, a segmentação oracional. Isso se justifica pelo fato de que, normalmente, uma oração corresponde a uma proposição simples no texto.

Como apresentado no capítulo anterior, na Subseção 3.2.3, Carlson e Marcu (2001) desenvolveram regras consistentes para a segmentação textual oracional. Diz-se que essas regras são consistentes pelo fato de poderem ser aplicadas uniformemente a diferentes textos de maneira coerente e não ambígua em um processamento automático. Para a segmentação do CorpusTCC, essas mesmas regras foram adotadas. Apesar de terem sido desenvolvidas para o inglês, elas são genéricas o suficiente para serem aplicadas ao português também.

4.1.2.3. Elenco de Relações Retóricas

Inicialmente, para a anotação do CorpusTCC, utilizou-se o conjunto de relações proposto originalmente pela RST (Mann e Thompson, 1987). Conforme a anotação progrediu, percebeu-se a necessidade de mais algumas relações, que foram, então, extraídas do trabalho de Marcu (1997), resultando no conjunto de 32 relações definidas no Apêndice A.

Como se discute no Capítulo 2, delimitaram-se, na anotação retórica, as orações relativas que expressam proposições por si só. Nestes casos, durante a anotação, as relações retóricas que relacionam as proposições expressas por estas orações a outras proposições têm em seu nome a extensão "-E", indicando que é uma relação "Encaixada" (em inglês, *Embedded*). Por exemplo, no trecho de texto abaixo, a proposição expressa pelo segmento (2) se relaciona com a proposição expressa pelo segmento (1) por uma relação ELABORATION-E. Como se pode perceber, o

segmento (2) fragmenta a proposição que é expressa pelos segmentos (1) e (3), devendo-se usar, neste caso, a relação estrutural SAME-UNIT para unificar esta proposição.

"(1) PLN, (2) que também é chamado Lingüística Computacional, (3) é uma grande área de pesquisa."

Apesar da diferenciação na nomenclatura, as relações encaixadas possuem o mesmo significado das relações tradicionais.

4.1.2.4. Estratégia de Anotação Retórica

Carlson e Marcu (2001) discutem as estratégias possíveis para a anotação retórica de textos. Com base na observação de seus anotadores, perceberam diferentes estratégias de anotação: alguns liam o texto antes, outros não; alguns, conforme segmentavam o texto, já relacionavam a proposição correspondente à estrutura retórica parcial já construída, realizando o que foi chamado de anotação incremental; outros estruturavam os parágrafos do texto isoladamente, para então integrá-los em uma única estrutura, realizando uma anotação modular.

Para a anotação do CorpusTCC, adotou-se a estratégia de anotação incremental e modular: primeiramente todas as proposições presentes em uma sentença foram relacionadas retoricamente; a seguir, todas as sentenças de um parágrafo foram relacionadas; por fim, os parágrafos foram relacionados. Essa estratégia se mostrou adequada e consistente para a anotação do corpus. Esse esquema de anotação se beneficia do fato de que o escritor tende a colocar juntas (isto é, no mesmo nível na hierarquia organizacional do texto) as informações relacionadas. Por exemplo, se duas proposições estão diretamente relacionadas, como uma causa e seu efeito, é provável que elas sejam expressas em uma única sentença ou em sentenças adjacentes.

Após a anotação do CorpusTCC estar completa, vários textos do corpus foram escolhidos aleatoriamente para terem sua anotação analisada. Em geral, não foram detectadas relações retóricas inadequadas. Em alguns casos, entretanto, notou-se que, em alguns contextos específicos, outras relações poderiam ser mais apropriadas.

Nesses casos, a anotação não sofreu alteração, pois a ambigüidade retórica é natural e, para que o corpus seja realmente representativo para sua finalidade, o fato dessa ambigüidade estar representada no corpus é desejável.

No geral, pode-se dizer que os relatos de Carlson e Marcu (2001) sobre os problemas e questões da anotação retórica foram evidenciados durante a anotação do CorpusTCC. Nesses relatos, feitos a partir da observação de anotadores durante a anotação, os autores discutem questões como: a preferência dos anotadores por certas relações em casos de ambigüidade; a mudança no julgamento do que é adequado (em termos da escolha de relações retóricas) conforme os anotadores têm mais experiência na anotação retórica; o tempo que os anotadores levam para anotar retoricamente textos conforme sua experiência no assunto aumenta.

Na Tabela 4.2, mostram-se o número de ocorrências e a freqüência das relações retóricas no CorpusTCC completamente anotado, não incluindo as relações encaixadas. Como se pode notar, algumas relações não ocorreram no corpus (por exemplo, JOINT) ou ocorreram poucas vezes (por exemplo, COMPARISON, OTHERWISE e SUMMARY). É natural que relações JOINT não tenham ocorrido no corpus, caso contrário, haveria quebra da coerência no texto (veja definição desta relação no Apêndice A). Como esperado, relações como ELABORATION e LIST foram as relações mais freqüentes, já que são as mais genéricas do elenco de relações utilizado. Marcu (1997) observou distribuições similares na marcação de seu corpus, apesar de ter utilizado um corpus de natureza diferente, composto por textos jornalísticos.

Na Tabela 4.3, mostram-se o número de ocorrências e a freqüência das relações retóricas encaixadas. Como se pode notar, muitas delas não ocorreram no corpus.

Tabela 4.2 – Número de ocorrências e freqüências das relações retóricas

Relações	Número de ocorrências	Freqüência (%)
ANTITHESIS	21	0,53
ATTRIBUTION	185	4,69
BACKGROUND	112	2,84
CIRCUMSTANCE	121	3,07
COMPARISON	5	0,13
CONCESSION	62	1,57
CONCLUSION	14	0,35
CONDITION	19	0,48
ELABORATION	1.030	26,10
ENABLEMENT	50	1,27
EXPLANATION	29	0,73
EVALUATION	10	0,25
EVIDENCE	15	0,38
INTERPRETATION	14	0,35
JUSTIFY	90	2,28
MEANS	48	1,22
MOTIVATION	18	0,46
NON-VOLITIONAL-CAUSE	64	1,62
NON-VOLITIONAL-RESULT	30	0,76
OTHERWISE	2	0,05
PARENTHETICAL	360	9,12
PURPOSE	318	8,06
RESTATEMENT	20	0,51
SOLUTIONHOOD	50	1,27
SUMMARY	4	0,10
VOLITIONAL-CAUSE	75	1,90
VOLITIONAL-RESULT	78	1,98
CONTRAST	89	2,26
JOINT	0	0
LIST	550	13,94
SAME-UNIT	393	9,96
SEQUENCE	70	1,77

Tabela 4.3 – Número de ocorrências e freqüências das relações encaixadas

Relações	Número de ocorrências	Freqüência (%)
ANTITHESIS-E	0	0
ATTRIBUTION-E	0	0
BACKGROUND-E	0	0
CIRCUMSTANCE-E	31	3,42
COMPARISON-E	6	0,66
CONCESSION-E	8	0,88
CONCLUSION-E	0	0
CONDITION-E	0	0
ELABORATION-E	651	71,85
ENABLEMENT-E	3	0,33
EXPLANATION-E	1	0,11
EVALUATION-E	5	0,55
EVIDENCE-E	0	0
INTERPRETATION-E	0	0
JUSTIFY-E	8	0,88
MEANS-E	18	1,99
MOTIVATION-E	1	0,11
NON-VOLITIONAL-CAUSE-E	2	0,22
NON-VOLITIONAL-RESULT-E	8	0,88
OTHERWISE-E	0	0
PURPOSE-E	139	15,34
RESTATEMENT-E	0	0
SOLUTIONHOOD-E	0	0
SUMMARY-E	0	0
VOLITIONAL-CAUSE-E	8	0,88
VOLITIONAL-RESULT-E	17	1,88

4.1.3. Extração de Conhecimento

Após estar completamente anotado, o CorpusTCC foi analisado manualmente em busca de marcadores textuais e das relações retóricas que estes sinalizam. Esta análise foi conduzida com o auxílio de uma ferramenta desenvolvida durante este trabalho de doutorado chamada RhetDB[3] (*Rhetorical DataBase*), pela qual é possível importar os dados produzidos pela *RST Annotation Tool* de Marcu e apresentá-los ao analista humano de forma amigável, oferecendo a este comandos avançados de busca por marcadores textuais e relações, além de campos de texto para anotação de informações que julgue importantes em sua análise, como os marcadores textuais associados a cada segmento textual. Grande parte dos dados lingüísticos produzidos

[3] A RhetDB foi implementada em Borland Delphi.

por este trabalho de doutorado foi extraída diretamente da base de dados desta ferramenta.

Como resultado da análise do CorpusTCC, foram produzidos padrões de análise para cada relação retórica, totalizando cerca de 740 padrões. Estes padrões constituem o principal repositório de informação do DiZer. Cada padrão de análise é composto por seis campos:

- nome da relação retórica sinalizada;
- a ordem entre as proposições relacionadas, isto é, se o núcleo é expresso antes do satélite no texto (NS) ou o oposto (SN) para relações mononucleares; no caso de relações multinucleares, dois núcleos são observados (NN);
- marcadores textuais no segmento que expressa a primeira proposição;
- posição dos marcadores anteriores no segmento que expressa a primeira proposição, caso estes existam;
- marcadores textuais no segmento que expressa a segunda proposição;
- posição dos marcadores anteriores no segmento que expressa a segunda proposição, caso estes existam.

Um marcador em um segmento pode estar localizado em seu "início", "meio" ou "fim", como se indica nos padrões de análise. Além disso, pode-se especificar que o marcador está fragmentado no segmento, em diversas posições, pela especificação "múltiplo".

Como exemplo, considere a estrutura retórica mostrada na Figura 4.2 extraída do CorpusTCC anotado. Como se pode ver, há uma relação retórica PURPOSE entre as proposições expressas pelos segmentos (1) e (2), com a segunda proposição sendo o satélite da relação. O padrão de análise produzido a partir desse texto é exibido na Figura 4.3, codificando o conhecimento de que a frase indicativa "tendo como objetivo" no início do segmento que expressa a segunda proposição sinaliza a relação PURPOSE, na qual o núcleo é expresso no texto antes do satélite (NS). Note que, no segmento que expressa a primeira proposição, não há marcador algum (indicado no padrão pela linha tracejada ---).

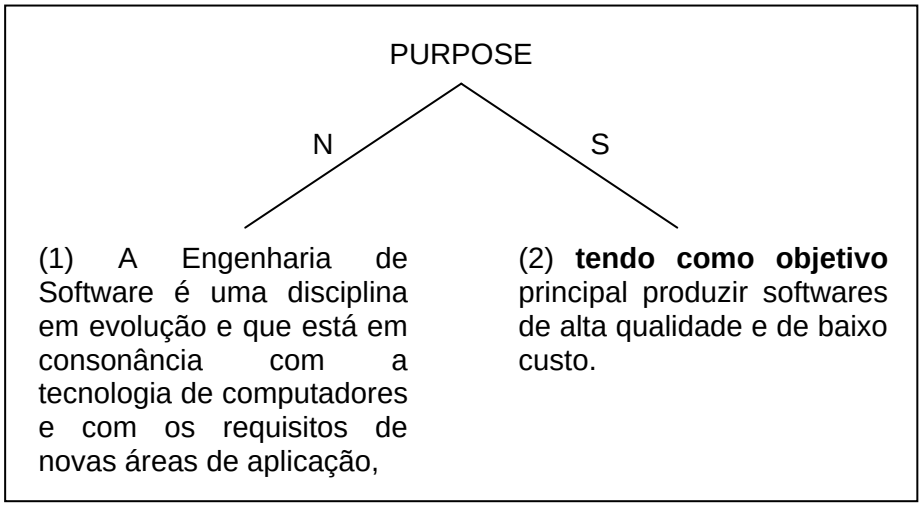


Figura 4.2 – Estrutura retórica para trecho de texto do CorpusTCC

Relação	PURPOSE
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	tendo como objetivo
Posição do segundo marcador	início

Figura 4.3 – Exemplo (1) de padrão de análise para a relação PURPOSE

Os padrões de análise podem codificar, também, informações morfossintáticas das palavras, suas formas canônicas (isto é, lemas) e/ou informações dependentes de gênero e domínio textual. Por exemplo, no padrão da Figura 4.4, tem-se uma relação retórica PURPOSE entre duas proposições quando, no segmento que expressa a segunda proposição, há uma palavra cuja forma canônica é "cujo", seguida por uma palavra da classe *palPur* (isto é, palavra que expressa objetivo – em inglês, *purpose*), que é seguida por uma palavra pertencente à classe dos adjetivos, que, por fim, é seguida por uma palavra cuja forma canônica é "ser". Um exemplo desse padrão é encontrado no trecho de texto "Este trabalho é baseado em diversas teorias discursivas, **cujos propósitos principais são** estruturar o discurso subjacente a um texto em seus vários níveis e representar as intenções do escritor em relação ao leitor". De acordo com o padrão de análise, "cujo" é a forma canônica da palavra "cujos", a palavra "propósitos" pertence à classe *palPur*, a palavra "principais" é um adjetivo e a palavra "ser" é a forma canônica do verbo "são". O requerimento da forma canônica no padrão é especificado pelo sufixo "_can" adicionado à palavra; a classe *palPur* contém todas as palavras que podem expressar um objetivo e é definida em um

repositório de informação a parte, como será discutido na próxima seção; a especificação morfossintática "_adj" informa que um adjetivo é requerido, independentemente de qual seja a palavra. As classes definidas a parte, como *palPur*, são, em geral, dependentes de gênero e domínio textual.

Relação	PURPOSE
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	cujo_can palPur _adj ser_can
Posição do segundo marcador	início

Figura 4.4 – Exemplo (2) de padrão de análise para a relação PURPOSE

Nos padrões de análise, podem-se especificar, ainda, pontos em que o casamento de padrões realizado pelo DiZer pode ignorar palavras em busca de um termo específico, ou seja, as componentes de um marcador textual não precisam estar adjacentes em um texto. Por exemplo, na Figura 4.5, pelo uso do símbolo * (asterisco), diz-se que entre a palavra pertencente à classe *palPur* e o adjetivo, pode haver um número qualquer de palavras. A especificação "múltiplo" para a posição deste marcador indica que suas partes estão dispersas no segmento que o contém. Esse tipo de representação atribui aos padrões de análise grande flexibilidade e permite que se representem marcadores textuais complexos e dependências de longa distância entre as palavras, quando isso se mostra necessário.

Relação	PURPOSE
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	cujo_can palPur * _adj ser_can
Posição do segundo marcador	múltiplo

Figura 4.5 – Exemplo (3) de padrão de análise para a relação PURPOSE

Em geral, cada marcador textual possível para cada relação foi codificado em um padrão de análise. Na Tabela 4.4, mostram-se o número e a porcentagem de relações retóricas marcadas superficialmente no CorpusTCC, não se distinguindo as relações encaixadas das não encaixadas. Por exemplo, a relação ANTITHESIS ocorreu 21 vezes (como se pode ver na Tabela 4.2), das quais, em apenas um caso, a relação não

foi marcada superficialmente. Com isso, tem-se que 95,2% das relações ANTITHESIS possuem algum marcador no CorpusTCC.

Na Tabela 4.5, mostra-se a distribuição de marcadores textuais entre núcleos e satélites das relações mononucleares. Por exemplo, para a relação CIRCUMSTANCE, o núcleo está marcado em 11,6% dos casos, o satélite em 80,4% dos casos e ambos (tanto o núcleo quanto o satélite) em apenas 8% dos casos. Na Tabela 4.6, mostra-se a distribuição de marcadores superficiais entre os núcleos das relações multinucleares.

Na Tabela 4.7, exibe-se a porcentagem de núcleos seguidos por satélites (NS) e satélites seguidos por núcleos (SN) para as relações retóricas mononucleares. Por exemplo, para a relação CONCESSION, o núcleo é realizado antes do satélite no CorpusTCC em 19,7% dos casos, enquanto o satélite é realizado antes do núcleo em 80,3% dos casos.

Tabela 4.4 – Porcentagem de relações marcadas superficialmente

Relação	Nro. rel. marcadas	% de rel. marcadas
ANTITHESIS	20	95,2
ATTRIBUTION	185	100
BACKGROUND	47	41,5
CAUSE	147	98,6
CIRCUMSTANCE	138	90,0
COMPARISON	11	100
CONCESSION	67	94,3
CONCLUSION	12	85,7
CONDITION	20	100
ELABORATION	1.010	60,0
ENABLEMENT	47	88,6
EVALUATION	14	93,3
EVIDENCE	3	20,0
EXPLANATION	23	76,6
INTERPRETATION	12	85,7
JUSTIFY	91	94,7
MEANS	60	90,9
MOTIVATION	16	84,2
OTHERWISE	2	100
PURPOSE	450	98,4
RESTATEMENT	17	85,0
RESULT	129	96,9
SOLUTIONHOOD	49	98,0
SUMMARY	4	100
CONTRAST	83	93,2
LIST	256	46,5
SEQUENCE	51	72,8

Tabela 4.5 – Porcentagem de núcleos e satélites marcados superficialmente para cada relação mononuclear

Relação	Somente núcleo marcado	Somente satélite marcado	Núcleo e satélite marcados
ANTITHESIS	85,0	15,0	0
ATTRIBUTION	0	100	0
BACKGROUND	76,6	8,5	14,9
CAUSE	45,6	24,4	30,0
CIRCUMSTANCE	11,6	80,4	8,0
COMPARISON	0	45,4	54,6
CONCESSION	35,8	56,7	7,5
CONCLUSION	0	100	0
CONDITION	0	90,0	10,0
ELABORATION	0	99,3	0,7
ENABLEMENT	83,0	14,9	2,1
EVALUATION	0	100	0
EVIDENCE	0	100	0
EXPLANATION	0	100	0
INTERPRETATION	0	100	0
JUSTIFY	8,8	9,9	81,3
MEANS	1,7	98,3	0
MOTIVATION	81,2	18,8	0
OTHERWISE	0	100	0
PURPOSE	0	97,3	2,7
RESTATEMENT	5,9	94,1	0
RESULT	3,9	93,8	2,3
SOLUTIONHOOD	0	4,1	95,9
SUMMARY	0	100	0

Tabela 4.6 – Distribuição de proposições marcadas para as relações multinucleares

Relação	Somente 1ª. proposição marcada (1º. núcleo)	Somente 2ª. proposição marcada (2º. núcleo)	1ª. e 2ª. proposições marcadas
CONTRAST	1,2	97,6	1,2
LIST	0,8	80,5	18,7
SEQUENCE	2,0	88,2	9,8

Tabela 4.7 – Porcentagem de núcleos seguidos por satélites (NS) e satélites seguidos por núcleos (SN) para as relações mononucleares

Relação	NS	SN
ANTITHESIS	14,2	85,8
ATTRIBUTION	2,7	97,3
BACKGROUND	0,9	99,1
CAUSE	24,8	75,2
CIRCUMSTANCE	49,3	50,7
COMPARISON	91,0	9,1
CONCESSION	19,7	80,3
CONCLUSION	100	0
CONDITION	50,0	50,0
ELABORATION	99,7	0,3
ENABLEMENT	24,5	75,5
EVALUATION	100	0
EVIDENCE	100	0
EXPLANATION	100	0
INTERPRETATION	100	0
JUSTIFY	78,1	21,9
MEANS	86,4	13,6
MOTIVATION	21,,0	79,0
OTHERWISE	100	0
PURPOSE	85,3	14,7
RESTATEMENT	95,0	5,0
RESULT	96,2	3,8
SOLUTIONHOOD	0	100
SUMMARY	100	0

Diante da importância dos marcadores discursivos para a análise retórica automática, mostra-se, na Tabela 4.8, a distribuição dos principais marcadores observados no CorpusTCC em função das relações retóricas que sinalizam. Na primeira linha da tabela, por exemplo, tem-se que o marcador “no entanto” foi observado 8 vezes com a relação ANTITHESIS no corpus. Note que algumas relações não possuem marcadores discursivos associados (ATTRIBUTION, por exemplo) (indicado na tabela pela linha tracejada ---).

Tabela 4.8 – Distribuição dos marcadores discursivos em função das relações que sinalizam

Relações retóricas	Marcadores discursivos
ANTITHESIS	apesar de (1) em paralelo (1) entretanto (5) mas (1) no entanto (8) porém (1)
ATTRIBUTION	---
BACKGROUND	após (2)

	assim (1) atualmente (3) dessa forma (1) desse modo (1) sendo assim (1)
CAUSE	assim (6) com (8) como (5) como conseqüência (2) dessa forma (3) desse modo (2) então (1) logo (1) nesse caso (1) pois (8) por causa (1) por esse motivo (2) por isso (5) porque (3) portanto (5) uma vez que (6)
CIRCUMSTANCE	a medida que (3) a partir de (1) antes (3) após (3) assim (2) atualmente (2) com (1) dessa forma (1) desse modo (1) em particular (1) nesse contexto (7) nesse sentido (3) onde (4) quando (45) visto isso (1)
COMPARISON	como (1)
CONCESSION	ainda (2) ao invés de (2) apesar de (14) contudo (1) em vez de (1) embora (10) entretanto (2) mas (3) mesmo assim (1) no entanto (8) porém (8)
CONCLUSION	assim (3) dessa forma (1) desse modo (3) portanto (2) sendo assim (1)
CONDITION	a medida que (1)

	caso (6) se (8)
CONTRAST	apesar de (1) contudo (1) em contrapartida (1) enquanto (9) entretanto (14) mas (14) no entanto (15) por outro lado (12) porém (14)
ELABORATION	adicionalmente (2) ainda (4) além de (58) bem como (2) como exemplo (6) como, por exemplo, (7) como também (3) dessa forma (1) em adição (2) em nível de (1) em particular (3) especificamente (3) inclusive (2) onde (28) por exemplo (19) principalmente (6) também (10)
ENABLEMENT	a partir de (1) assim (1) através (2) com (3)
EVALUATION	---
EVIDENCE	---
EXPLANATION	isto é (1) pois (12) porque (1)
INTERPRETATION	assim sendo (1) dessa forma (1) dessa maneira (1) isto é (2) ou seja (4) sendo assim (1)
JUSTIFY	assim (1) como (1) logo (1) nesse caso (1) nesse contexto (1) pois (17) porque (2) uma vez que (7) visto isso (1)
LIST	além de (3) ao mesmo tempo (3)

	ao passo que (2) bem como (2) como também (1) e (133) em geral (2) enquanto (11) finalmente (2) já (2) mais tarde (1) ou (15) quando (1) também (3)
MEANS	através (9) para esse fim (1) para isso (7) para tanto (3)
MOTIVATION	diante de (2)
OTHERWISE	caso contrário (1) ou, alternativamente, (1)
PURPOSE	a fim de (11) para (198)
RESTATEMENT	assim (2) dessa forma (1) isto é (4) ou seja (10)
RESULT	aí (1) assim (6) com (4) conseqüentemente (1) dessa forma (6) desse modo (1) dessa maneira (1) logo (1) nesse caso (2) por isso (1) por causa (1) porque (1) portanto (3) uma vez que (1)
SEQUENCE	ainda (1) a partir de (6) a seguir (2) após (2) atualmente (3) depois (3) diante de (1) e (16) em sequida (3) então (1) finalmente (2) logo após (1) ou (1) por fim (1) posteriormente (1)

	primeiramente (1)
SOLUTIONHOOD	---
SUMMARY	ou seja (1) resumindo (2)

No Apêndice B, os marcadores textuais representativos encontrados no CorpusTCC para cada relação retórica são exibidos juntamente com exemplos. Em geral, todos esses marcadores e as relações que sinalizam foram codificados em padrões de análise.

Para as relações que não possuem marcadores textuais, foi possível desenvolver heurísticas para identificá-las em função das palavras indicativas de tais relações detectadas durante a análise de corpus. Isso aconteceu para as relações retóricas EVALUATION e SOLUTIONHOOD, cujas heurísticas são mostradas nas Figuras 4.6 e 4.7, respectivamente.

Se, em um segmento X, palavras de cunho avaliativo, como “adequado” e “sucesso”, aparecem mais de uma vez, então uma relação EVALUATION deve ser estabelecida entre as proposições expressas pelos segmentos X e Y, o qual antecede X no texto, com a proposição expressa por X sendo o satélite da relação.

Figura 4.6 – Heurística para identificação da relação retórica EVALUATION

Se, em um segmento X, palavras de cunho negativo, como “custo” e “problema”, aparecem mais de uma vez e, em um segmento Y que segue X, palavras de cunho positivo, como “solução” e “desenvolvimento”, aparecem mais de uma vez também, então uma relação SOLUTIONHOOD deve ser estabelecida entre as proposições expressas pelos segmentos X e Y, com a proposição expressa por X sendo o satélite da relação.

Figura 4.7 – Heurística para identificação da relação retórica SOLUTIONHOOD

Por exemplo, no trecho de texto da Figura 4.8, há uma relação EVALUATION entre as proposições expressas pelos segmentos (1) e (2). As palavras indicativas da relação são mostradas em negrito.

(1) Um sistema baseado em aeromodelos e equipamentos de rádio controle convencionais foi desenvolvido para a avaliação do emprego da tecnologia no monitoramento de problemas agrícolas. (2) Os resultados obtidos são altamente **adequados** à utilização do sistema em várias aplicações, mostrando o **sucesso** do sistema e sua **flexibilidade** para diversas tarefas.

Figura 4.8 – Trecho de texto com relação EVALUATION

Assim como as palavras da classe *palPur*, que expressam objetivo, as palavras de cunhos "avaliativo", "negativo" e "positivo" pertencem a classes especificadas a parte em um repositório do DiZer, como será discutido a seguir.

4.2. Arquitetura do DiZer

A arquitetura do DiZer[4] é exibida na Figura 4.9.

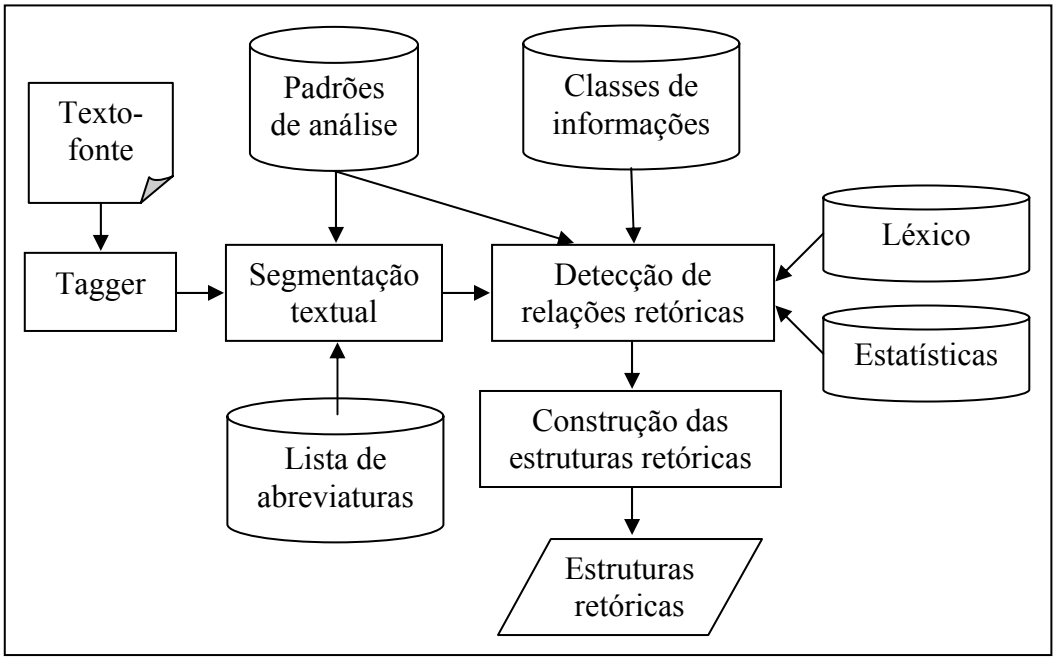


Figura 4.9 – Arquitetura do DiZer

De acordo com esta arquitetura, um texto-fonte é dado como entrada para o sistema e os seguintes passos são realizados para sua anotação retórica:

[4] A interface gráfica do DiZer foi implementada em Borland Delphi, enquanto seu engenho de inferência em Amzi! Prolog.

1. inicialmente, o texto-fonte é etiquetado morfossintaticamente, ou seja, as classes gramaticais das palavras são especificadas;
2. o texto etiquetado é segmentado;
3. todas as relações retóricas possíveis entre as proposições expressas pelos segmentos do texto delimitados no passo anterior são detectadas;
4. com base nas relações entre as proposições detectadas, as estruturas retóricas possíveis para o texto-fonte são construídas.

O tagger utilizado é o MXPOST (Ratnaparkhi, 1996) treinado para o português do Brasil (Aires et al., 2000). Esse etiquetador utiliza vinte etiquetas que representam, basicamente, as classes gramaticais tradicionais do português, possuindo precisão geral de 89%.

Nas próximas subseções, os repositórios de informação utilizados no DiZer e seus processos são descritos.

4.2.1. Repositórios de Informação do DiZer

Os repositórios de padrões de análise e de classes de informações utilizados no DiZer foram definidos a partir da análise do CorpusTCC, como discutido anteriormente.

O repositório de padrões de análise contém, aproximadamente, 740 padrões. O repositório de classes de informações contém as classes utilizadas nos padrões de análise, por exemplo, a classe *palPur* discutida anteriormente, que contém todas as palavras (em suas formas canônicas) que expressam objetivo (que são "objetivo", "propósito", "intenção", "intuito", "perspectiva", "tentativa" e "função"). Este repositório é, em geral, dependente do gênero e do domínio do texto sob análise, pois as classes especificadas são utilizadas, em sua maioria, para compor as palavras e frases indicativas dos padrões de análise, as quais costumam variar com o tipo de texto sendo analisado.

A lista de abreviaturas e o léxico utilizados no DiZer são provenientes do corretor gramatical ReGra (Martins et al., 1998) para o português do Brasil. A lista de abreviaturas é abrangente, contendo, aproximadamente, 240 abreviaturas, e o léxico é o maior léxico computacional para a língua portuguesa, com mais de 1.500.000 palavras e suas correspondentes formas canônicas. Estes repositórios são utilizados

nos processos de segmentação textual e detecção de relações retóricas, como será explicado.

O repositório de estatísticas contém dados probabilísticos sobre a organização discursiva de textos científicos. Esses dados foram coletados automaticamente do CorpusTCC anotado e são utilizados para ranquear as estruturas retóricas produzidas pelo DiZer, nos casos em que mais de uma estrutura é produzida.

Mais especificamente, o repositório de estatísticas contém a probabilidade de se observarem as relações retóricas com as proposições ou relações raízes das subestruturas que conectam. Por exemplo, a probabilidade de se observar uma relação retórica CAUSE se estabelecendo entre uma subestrutura retórica com a relação LIST como raiz e uma proposição, com a primeira sendo o núcleo da relação, é de 1,3%, como especificado abaixo:

P(LIST,N,proposição,S|CAUSE)=0,013

De forma similar, a probabilidade de se observar uma relação LIST se estabelecendo entre duas proposições, ambas nucleares, é de 25,3%.

P(proposição,N,proposição,N|LIST)=0,253

Basicamente, a probabilidade utilizada é a probabilidade de se ter um nó intermediário em uma estrutura retórica observado com seus nós filho correspondentes e a nuclearidade deles.

As probabilidades foram coletadas do corpus da seguinte forma: para uma relação R conectando subestruturas ou proposições X e Y, supondo-se, por exemplo, que a primeira é núcleo e a segunda satélite da relação, a probabilidade P(X,N,Y,S|R) é obtida pela aplicação da seguinte fórmula:

$$P(X,N,Y,S \mid R) = \frac{Nro.\ de\ subestruturas\ rhet\_rel(R,Y,X)}{Nro.\ de\ subestruturas\ rhet\_rel(R,\_,\_)\ com\ quaisquer\ N\ e\ S}$$

ou seja, a probabilidade P(X,N,Y,S|R) é a relação entre o número de vezes que tal subestrutura é observada e o número de subestruturas relacionadas pela relação R.

4.2.2. Processos do DiZer

4.2.2.1. Segmentação Textual

O processo de segmentação textual recebe como dado de entrada o texto-fonte anotado morfossintaticamente. Sua função é identificar as orações do texto, pois, normalmente, elas expressam proposições simples, a unidade básica de uma estrutura retórica.

Por meio de regras simples baseadas na ocorrência dos sinais de pontuação tradicionais (por exemplo, ponto final, ponto de exclamação e ponto de interrogação), delimitam-se as sentenças do texto. Devido à ambigüidade do ponto, que pode indicar ou não final de sentença (se for o ponto de uma abreviatura, por exemplo, ele pode não indicar final de sentença), consulta-se a lista de abreviaturas para se decidir se a sentença em questão deve ou não ser delimitada.

Para delimitação das orações dentro de uma sentença, (a) procuram-se nesta marcadores textuais "fortes" em seu interior (isto é, marcadores que claramente sinalizam relações retóricas) ou sinais de vírgula, ponto e vírgula e dois pontos, entre outros, e (b) verifica-se, por meio das etiquetas morfossintáticas associadas às palavras, se cada um dos segmentos possíveis nos dois lados do marcador ou sinal identificado contém pelo menos um verbo: se isso ocorrer, então os segmentos são delimitados; caso contrário, a sentença não é segmentada. A busca por um verbo nos segmentos dentro da sentença consiste em uma heurística simples que garante que segmentos sem verbos não sejam delimitados. Apesar de um segmento sem verbo poder expressar uma proposição, a estratégia adotada garante uma segmentação automática mais consistente e livre de erros, como discutido no manual de segmentação desenvolvido por Carlson e Marcu (2001).

Como exemplo, considere a sentença extraída do CorpusTCC "Será utilizado também o termo Tolerância a Defeitos referindo-se a *Fault Tolerance*, embora usualmente este seja empregado como Tolerância a Falhas.". Nesta sentença, além do sinal de vírgula em seu interior como indicador de possível início de um novo segmento, há o marcador discursivo "embora", o qual claramente sinaliza uma relação retórica de oposição (CONCESSION, neste caso). Por outro lado, na sentença "O significado é sempre entendido pelos membros das organizações e muitos dos dados

mantidos em ambas as organizações têm o mesmo significado.", não se segmenta a sentença na posição em que se encontra o marcador discursivo "e", pois este é um marcador fraco, isto é, é ambíguo e não sinaliza claramente uma relação retórica, podendo ser utilizado com outras funções, por exemplo, como marcador sentencial ou pragmático (veja Capítulo 3).

Os marcadores textuais buscados no interior de uma sentença para sua possível segmentação são os marcadores especificados nos padrões de análise armazenados no repositório de padrões, como mostrado na Figura 4.9. Durante o processo de segmentação, os padrões são analisados pelo DiZer de forma exaustiva, até que se encontre um marcador textual presente na sentença sendo processada.

Opcionalmente, no DiZer, pode-se realizar segmentação sentencial no texto, em vez de oracional. Com isso, considera-se que as proposições que serão os nós terminais na estrutura retórica a ser produzida não são, necessariamente, proposições simples, já que uma sentença pode expressar mais do que um simples fato ou evento.

Por fim, ao segmentar o texto, o DiZer armazena quais segmentos representam fins de parágrafo no texto. Isso é feito pela simples verificação da presença dos símbolos que indicam uma nova linha no texto. Essa informação é necessária ao DiZer devido a forma como o processo de detecção de relações retóricas é realizado, o qual é incremental e modular, de forma similar à realização da anotação retórica do CorpusTCC. Esse processo é apresentado na próxima subseção.

Como exemplo do processo de segmentação, na Figura 4.10, mostra-se um texto segmentado pelo DiZer. Seis segmentos foram delimitados pelo sistema.

[Desde a sua abertura comercial, em 1993, a Internet tornou-se um meio de comunicação poderoso,]$_1$ [ao permitir a um usuário entrar em contato com quaisquer outros, espalhados pelo mundo todo.]$_2$

[O comércio eletrônico é um dos novos nichos de exploração comercial da rede mundial de computadores,]$_3$ [pois ela torna possível realizar transações comerciais de forma global, com custo de manutenção inferior ao empregado em uma rede de comércio tradicional.]$_4$

[O objetivo deste trabalho é apresentar uma proposta para o projeto e implementação de um serviço de comércio eletrônico na plataforma JAMP.]$_5$ [Esta plataforma constitui-se em um middleware implementado em Java/RMI para desenvolvimento de aplicações multimídia distribuídas, e em particular, aplicações para World Wide Web (WWW), através de frameworks de serviços para suporte ao desenvolvimento destas aplicações.]$_6$

Figura 4.10 – Texto segmentado pelo DiZer

As dificuldades encontradas no processo de segmentação residem na identificação das orações relativas, que, usualmente, expressam proposições simples. No geral, o DiZer não é capaz de delimitar corretamente tais segmentos. O uso de um parser poderia resolver esse problema, entretanto, até o presente momento, não há para o português do Brasil um parser de uso livre e suficientemente robusto para lidar com as sentenças sofisticadas de textos científicos.

4.2.2.2. Detecção das Relações Retóricas

Neste processo, a partir dos segmentos delimitados no processo de segmentação textual, procuram-se por todas as possíveis relações retóricas entre as proposições expressas por estes segmentos. Para tanto, realiza-se um processo de casamento de padrões, como definido por Russel e Norvig (2003), entre os padrões de análise armazenados no repositório de padrões e os possíveis pares de segmentos identificados pelo DiZer, consultando-se, quando necessário, o repositório de classes.

Um processo de casamento de padrões consiste em verificar se um par de segmentos contém as informações especificadas em um padrão de análise. Se isso ocorrer, estabelece-se a relação indicada no padrão entre as proposições expressas pelos segmentos, obedecendo-se a ordenação entre o núcleo e o satélite (ou outro núcleo, no caso de uma relação multinuclear) especificado no padrão.

O processo de casamento de padrões para dois segmentos quaisquer é aplicado exaustivamente para todos os padrões de análise, produzindo todas as relações retóricas possíveis entre as proposições expressas pelos segmentos. Por exemplo, para o par de segmentos 3 e 4 da Figura 4.10, as relações retóricas CAUSE, EXPLANATION e JUSTIFY são listadas como possíveis, sendo que a proposição expressa pelo segmento 3 é o núcleo destas relações. Os padrões aplicados para a determinação destas três relações são mostrados nas Figuras 4.11-4.13.

Relação	CAUSE
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	pois
Posição do segundo marcador	início

Figura 4.11 – Padrão de análise aplicado pelo DiZer para a relação CAUSE

Relação	EXPLANATION
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	pois
Posição do segundo marcador	início

Figura 4.12 – Padrão de análise aplicado pelo DiZer para a relação EXPLANATION

Relação	JUSTIFY
Ordem	NS
Marcador na 1a. proposição	---
Posição do primeiro marcador	---
Marcador na 2a. proposição	pois
Posição do segundo marcador	início

Figura 4.13 – Padrão de análise aplicado pelo DiZer para a relação JUSTIFY

Note que os três padrões especificam o mesmo marcador discursivo no início de um segmento para que as relações sejam estabelecidas.

Nos casos em que os padrões de análise fazem uso de informações sobre classes de palavras, o repositório de classes é consultado. Quando os marcadores textuais se referem às formas canônicas das palavras, o léxico acoplado ao DiZer é consultado durante o processo de casamento de padrões para se verificar se as palavras presentes no segmento textual possuem as formas canônicas correspondentes requeridas.

É importante dizer que, na análise automática realizada pelo DiZer, não se diferenciam as relações volitivas das não volitivas para as relações CAUSE e RESULT, como se pode notar no padrão de análise da Figura 4.11. Isso se deve ao fato de estas relações possuírem os mesmos marcadores textuais, não sendo possível, portanto, distingui-las.

Quando nenhum padrão se aplica para dois segmentos quaisquer, a relação ELABORATION é utilizada para relacionar suas proposições, pois esta é a relação mais freqüente no CorpusTCC (veja Seção 4.1 neste capítulo) e a mais genérica

definida pela RST. O fato de nenhum padrão se aplicar pode se dever à ausência de marcadores textuais nos segmentos ou à presença de marcadores que não constam no CorpusTCC e, portanto, não possuem padrões de análise correspondentes. Quando nenhum padrão se aplica, a inclusão da relação ELABORATION pode causar a construção de estruturas retóricas inadequadas, entretanto, tal escolha parece ser a mais natural nestes casos.

De forma similar à anotação retórica do CorpusTCC, o DiZer utiliza a estratégia modular e incremental, da esquerda para a direita, para detecção das relações retóricas entre as proposições expressas pelos segmentos: inicialmente, relacionam-se as proposições expressas pelas orações dentro de cada sentença; a seguir, relacionam-se as proposições expressas pelas sentenças dentro de cada parágrafo; por fim, relacionam-se as proposições expressas pelos parágrafos do texto. Como já discutido, tal estratégia se beneficia do fato de o escritor do texto expressar as proposições relacionadas em um mesmo nível de organização hierárquica no texto. Em termos práticos, esta estratégia diminui significativamente as possibilidades de análise pelo DiZer, tornando o processamento automático mais eficiente.

Na Figura 4.14, mostram-se todas as relações encontradas pelo DiZer para os segmentos mostrados na Figura 4.10. Na representação empregada, o predicado rhet_rel(R,Y,X) representa uma estrutura retórica na qual a proposição (ou subestrutura) Y é o satélite e a proposição (ou subestrutura) X o núcleo na relação retórica R.

```
rhet_rel(CIRCUMSTANCE, 2, 1)
rhet_rel(ENABLEMENT, 1, 2)
rhet_rel(CAUSE, 4, 3)
rhet_rel(EXPLANATION, 4, 3)
rhet_rel(JUSTIFY, 4, 3)
rhet_rel(ELABORATION, 6, 5)
rhet_rel(ELABORATION, [1, 2], [3, 4])
rhet_rel(BACKGROUND, [3, 4], [5, 6])
rhet  rel(PURPOSE, [5, 6], [3, 4])
```

Figura 4.14 – Relações retóricas detectadas pelo DiZer

A partir do conjunto de relações retóricas produzido pelo DiZer, constroem-se as estruturas retóricas possíveis para o texto, como se explica na subseção seguinte.

As limitações do processo de detecção de relações retóricas como realizado pelo DiZer são que, (a) em teoria, as relações do discurso não se estabelecem, necessariamente, entre proposições expressas por segmentos adjacentes no texto e (b) é comum que segmentos textuais não possuam marcadores textuais. Em relação a (a), Marcu (1997) mostra que é possível construir estruturas igualmente adequadas nas quais as relações se estabelecem somente entre proposições expressas por segmentos adjacentes, utilizando-se, para isso, o critério da composicionalidade; quanto a (b), na análise de um texto pelo DiZer, pode-se causar a produção de uma grande quantidade de relações ELABORATION.

4.2.2.3. Construção das Estruturas Retóricas

Neste processo, as estruturas retóricas possíveis para um texto são produzidas a partir das relações detectadas entre suas proposições no processo anterior. Utiliza-se, para isso, o algoritmo proposto por Marcu (1997), apresentado na Subseção 3.2.1.4 do Capítulo 3.

Como já explicado, o algoritmo de Marcu produz uma gramática na forma normal de Chomsky que codifica todas as possibilidades de relacionamento entre as proposições consideradas. Ao ser executada, esta gramática gera todas as estruturas retóricas possíveis para o texto sob análise.

O algoritmo proposto baseia-se no critério de composicionalidade, segundo o qual é possível relacionar duas proposições/subestruturas retóricas por uma relação qualquer quando esta se estabelece entre os núcleos das proposições/subestruturas retóricas. Esta restrição garante que somente estruturas retóricas formalmente válidas (segundo os princípios da RST) sejam construídas. Por outro lado, por ser uma restrição muito rígida, o critério de composicionalidade pode impedir a construção de qualquer estrutura por um sistema de análise automática, dado que este pode não ter precisão alta o suficiente para detectar as relações retóricas e a nuclearidade das proposições adequadamente. Diante disto, no DiZer, flexibiliza-se este critério quando isso se mostra necessário, anulando-se sua aplicação. Essa decisão possibilita a construção de estruturas retóricas para qualquer texto, mesmo que estas estruturas sejam completa ou parcialmente inadequadas.

Para o conjunto de relações da Figura 4.14, a estrutura da Figura 4.15 é produzida. Neste caso, o critério da composicionalidade não foi aplicado. Apesar de ser uma estrutura formalmente inválida (note, por exemplo, que, segundo o conjunto de relações da Figura 4.14, a relação BACKGROUND não se estabelece entre as proposições mais nucleares das subestruturas que conecta na Figura 4.15), pode-se verificar que ela é plausível.

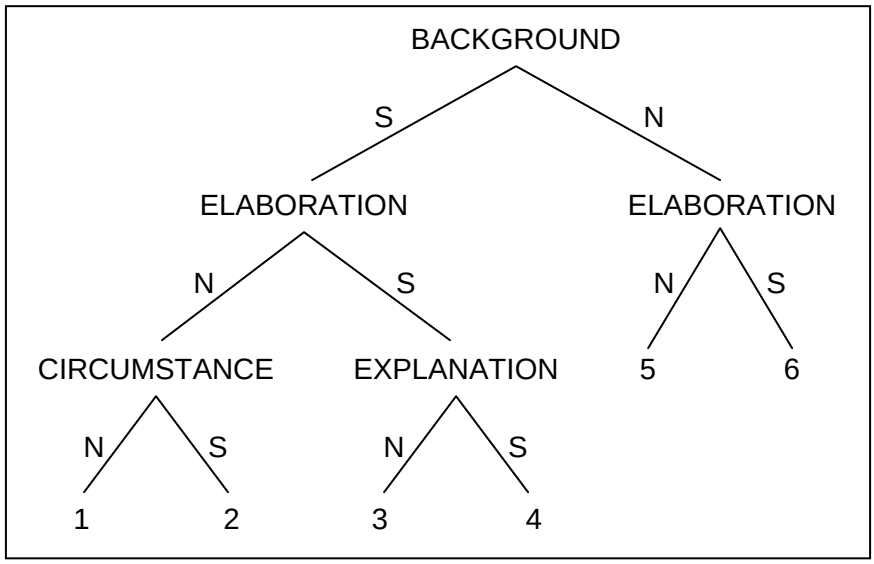


Figura 4.15 – Estrutura retórica construída pelo DiZer

O algoritmo de Marcu, com ou sem o uso do critério da composicionalidade, produz todas as estruturas retóricas possíveis para um texto. Em geral, várias estruturas são produzidas e, quanto maior o texto, mais estruturas existirão, pois há mais proposições para se relacionar, aumentando o número de possibilidades de construção de estruturas. Em algumas situações, pode-se desejar ter somente uma estrutura, a mais plausível ou provável, por exemplo, na aplicação do DiZer em outras tarefas de PLN, como sumarização automática ou resolução anafórica. Para tanto, no DiZer, há a possibilidade de se ranquear as estruturas produzidas de acordo com suas probabilidades.

A probabilidade associada a cada estrutura é obtida pelo uso das estatísticas armazenadas no repositório de estatísticas (veja Figura 4.9), descrito na Subseção 4.2.1. Para se calcular a probabilidade de uma estrutura retórica produzida, multiplicam-se as probabilidades das subestruturas que a compõem. Quando alguma probabilidade não é encontrada, devido à observação de subestruturas que não

constam no CorpusTCC anotado que deu origem ao repositório de estatísticas, utiliza-se como probabilidade o valor mínimo 10^{-6}. Como exemplo, a probabilidade da estrutura da Figura 4.15 é calculada da seguinte forma pelo DiZer:

P(estrutura retórica) =
P(ELABORATION,S,ELABORATION,N|BACKGROUND) x
P(CIRCUMSTANCE,N,EXPLANATION,S|ELABORATION) x
P(proposição,N,proposição,S|ELABORATION) x
P(proposição,N,proposição,S|CIRCUMSTANCE) x
P(proposição,N,proposição,S|EXPLANATION)

A seguir, relata-se a avaliação do DiZer.

4.3. Avaliação do DiZer

Para a avaliação do desempenho do DiZer em produzir estruturas retóricas plausíveis, foi desenvolvido um corpus de referência anotado retoricamente segundo a RST chamado RHETALHO. Diz-se que ele é um corpus de referência pelo fato das estruturas retóricas produzidas para seus textos serem o resultado da concordância entre pelo menos 2 anotadores humanos especialistas em RST (um deles é o autor deste trabalho) e poderem, portanto, ser utilizadas para comparação com estruturas produzidas automaticamente.

Em seu estágio atual, o RHETALHO possui cerca de 50 textos, 30 do gênero científico do domínio da Computação, contendo introduções e conclusões de textos, e 20 textos jornalísticos das Seções Cotidiano, Mundo e Ciências do jornal *on-line* Folha de São Paulo.

Para a construção do RHETALHO, os anotadores humanos utilizaram o mesmo conjunto de relações utilizado pelo DiZer e, para que houvesse concordância nas anotações, especialmente na determinação das relações retóricas, seguiram um protocolo de anotação. Para a segmentação textual, as regras do manual de segmentação de Carlson e Marcu (2001) foram utilizadas. Os textos foram anotados com a ferramenta de edição gráfica *RST Annotation Tool* de Marcu. O protocolo de anotação utilizado pelos anotadores é mostrado no Apêndice C.

O uso do RHETALHO possibilitou a avaliação do desempenho do DiZer de forma clara e objetiva, sem a introdução da subjetividade humana. Foram utilizadas, na avaliação, as 20 introduções de textos científicos da Computação contidas no RHETALHO. As estruturas possíveis para cada texto foram produzidas automaticamente pelo DiZer e comparadas com as estruturas previstas no corpus.

Nessa avaliação, verificou-se o desempenho do sistema na realização de suas três principais tarefas: segmentação textual, determinação da nuclearidade das proposições e detecção das relações retóricas entre elas. Para todos os casos, foram calculadas as medidas de cobertura e precisão tradicionais:

- para a segmentação textual, cobertura indica a relação entre os segmentos corretamente delimitados em relação a tudo que deveria ser delimitado (como especificado no RHETALHO) e precisão indica a relação entre os segmentos corretamente delimitados em relação a tudo que foi delimitado pelo sistema;
- para a determinação da nuclearidade das proposições, cobertura indica a relação entre as proposições corretamente classificadas (como núcleo ou satélite) em relação a tudo que deveria ser classificado (como especificado no RHETALHO) e precisão indica a relação entre as proposições corretamente classificadas em relação a tudo que foi classificado pelo sistema;
- para a detecção das relações retóricas entre as proposições, cobertura indica a relação entre as relações retóricas entre proposições corretamente detectadas em relação a tudo que deveria ser detectado (como especificado no RHETALHO) e precisão indica a relação entre relações retóricas entre proposições corretamente detectadas em relação a tudo que foi detectado pelo sistema.

Calculou-se, também, para cada item avaliado, a *F-Measure*, que é uma medida única do desempenho do sistema que combina a cobertura (C) e a precisão (P), indicando o quão próximo do ideal o sistema está. A medida *F-Measure* é calculada pela fórmula abaixo (seu resultado se estabelece em uma escala de 0 a 100%):

$$F - Measure = \frac{2 \times C \times P}{C + P}$$

Para fins de comparação e validação dos resultados do DiZer, também foi avaliado um método *baseline* de análise retórica automática. Por método *baseline*, refere-se a

um método cujos resultados sirvam de referência para que se julgue a qualidade dos resultados do DiZer. Este método realiza segmentação sentencial do texto, sem procurar por orações, e determina somente relações ELABORATION (pois esta é a relação mais genérica e freqüente) entre as proposições expressas pelas sentenças delimitadas, com a primeira proposição sendo o núcleo da relação. A estratégia modular e incremental de análise também é utilizada nesse método, assim como se faz no DiZer.

Nas Tabelas 4.9 e 4.10, mostram-se os resultados do DiZer e do método *baseline* para as tarefas de segmentação textual, determinação da nuclearidade das proposições e detecção das relações retóricas, considerando-se segmentação oracional e sentencial, respectivamente. Mostra-se o desempenho do sistema em termos da Coburtura (C), Precisão (P) e F-Measure (F) médios.

Como se pode ver pelas tabelas, o DiZer superou o método *baseline* nos dois casos, com uma grande diferença quando a segmentação oracional é realizada, como esperado. Isso se deve ao fato da segmentação oracional produzir estruturas retóricas com granularidade mais fina, que se aproximam mais das estruturas retóricas do RHETALHO.

Tabela 4.9 – Desempenho do DiZer para segmentação sentencial com textos científicos

	DiZer (%)			**Método *baseline* (%)**		
Tarefas	C	P	F	C	P	F
Segmentação textual	25,2	41,7	31,4	25,2	41,7	31,4
Determinação da nuclearidade	39,1	69,5	50,1	32,4	59,5	42,0
Detecção de relações	28,7	61,0	39,1	20,7	49,2	29,2

Tabela 4.10 – Desempenho do DiZer para segmentação oracional com textos científicos

	DiZer (%)			**Método *baseline* (%)**		
Tarefas	C	P	F	C	P	F
Segmentação textual	57,3	56,2	56,8	25,2	41,7	31,4
Determinação da nuclearidade	79,7	82,3	80,9	32,4	59,5	42,0
Detecção de relações	63,2	61,9	62,5	20,7	49,2	29,2

Em relação à aplicação do critério de composicionalidade durante a execução do algoritmo de Marcu, verificou-se que, para a realização de segmentação sentencial, o critério pôde ser aplicado em 75% dos casos pelo DiZer. Para a realização da segmentação oracional, o critério foi aplicado em apenas 20% dos casos.

Para verificar a possibilidade de utilização do DiZer com textos de outros gêneros e domínios, o sistema também foi avaliado com 5 textos jornalísticos da Seção Mundo do jornal *on-line* Folha de São Paulo contidos no RHETALHO. A avaliação foi conduzida de forma idêntica à anterior e os resultados são mostrados nas Tabelas 4.11 e 4.12 para a realização de segmentação sentencial e oracional pelo DiZer, respectivamente.

Apesar do desempenho inferior ao desempenho com textos científicos, o DiZer superou o método *baseline* para a realização da segmentação oracional. Para a segmentação sentencial, entretanto, o método *baseline* apresentou um desempenho melhor. Acredita-se que isso se deve à forma como os textos jornalísticos são organizados: em geral, a maioria das relações observadas são relações ELABORATION, com a primeira proposição da relação sendo o núcleo. Esta é exatamente a estratégia de análise utilizada pelo método *baseline*. Quando se realiza a segmentação oracional, entretanto, o DiZer possui um desempenho melhor do que o método *baseline* porque é capaz de produzir estruturas mais próximas das estruturas previstas no RHETALHO.

Tabela 4.11 – Desempenho do DiZer para segmentação sentencial com textos jornalísticos

	DiZer (%)			**Método *baseline* (%)**		
Tarefas	C	P	F	C	P	F
Segmentação textual	9,9	20,6	13,4	9,9	20,6	13,4
Determinação da nuclearidade	22,3	55,3	31,8	28,4	71,3	40,7
Detecção de relações	12,5	38,3	18,9	17,6	58,3	27,0

Tabela 4.12 – Desempenho do DiZer para segmentação oracional com textos jornalísticos

	DiZer (%)			**Método *baseline* (%)**		
Tarefas	C	P	F	C	P	F
Segmentação textual	48,8	54,1	51,3	9,9	20,6	13,4
Determinação da nuclearidade	55,8	63,5	59,4	28,4	71,3	40,7
Detecção de relações	37,8	43,2	40,3	17,6	58,3	27,0

Sobre a aplicação do critério de composicionalidade, verificou-se que, para a realização da segmentação sentencial, o critério pôde ser aplicado em 60% dos casos pelo DiZer. Para a segmentação oracional, o critério foi aplicado em apenas 20% dos casos, o mesmo índice conseguido para a avaliação com textos científicos.

Considera-se natural o desempenho inferior do DiZer ao ser avaliado com textos jornalísticos em relação à avaliação com textos científicos, pois o DiZer foi desenvolvido a partir da análise de um corpus de textos científicos. As avaliações apresentadas mostram que os resultados do DiZer são satisfatórios, validando sua metodologia de desenvolvimento e possibilitando o uso do sistema em outras aplicações de PLN. Logicamente, há melhorias que podem ser feitas, como se discute no Capítulo 6, as quais constituem alguns dos trabalhos futuros relativos a esta tese de doutorado.

Pela avaliação mostrada, o DiZer se encontra em um patamar próximo dos analisadores retóricos similares, cujo maior representante é o de Marcu (1997, 2000b). Apesar das diferenças evidentes entre os analisadores (como as línguas para as quais foram desenvolvidos, os tipos de textos que analisam, as metodologias de avaliação e os próprios corpora utilizados na avaliação) que inviabilizam uma comparação justa, tal comparação possibilita vislumbrar os resultados que se atingem com sistemas desse tipo. Como mostrado no Capítulo 3, o parser retórico de Marcu apresenta os seguintes resultados: para a tarefa de segmentação textual, precisão de 90% e cobertura de 81%; para a tarefa de determinação da nuclearidade das proposições, precisão de 85% e cobertura de 50%; para a tarefa de detecção de relações retóricas, precisão de 78% e cobertura de 47%. Os resultados do DiZer que mais se aproximam da forma como os resultados de Marcu foram obtidos são os mostrados na Tabela 4.10.

No próximo capítulo, descrevem-se os modelos estatísticos desenvolvidos para a realização da análise discursiva automática.

5. Modelos Estatísticos para Análise Discursiva Automática

Apresentam-se, neste capítulo, modelos estatísticos inéditos desenvolvidos para a realização de análise discursiva automática. Por modelo, refere-se a construtos que tentam explicar como eventos/processos do mundo real ocorrem. Por modelo estatístico, ou probabilístico, entende-se um modelo que faz uso de probabilidades para determinar como e que eventos/processos ocorrem (Manning e Schütze, 1999). Modelo estatístico para análise discursiva significa, portanto, um modelo que simula o processo de análise discursiva por meio de probabilidades.

Os modelos desenvolvidos são baseados no modelo *Noisy-Channel* de Shannon (1948) e treinados por meio do método de Aprendizado de Máquina *Expectation-Maximization* – EM (Dempster et al., 1977). Esses modelos visam a aprender regras semânticas, modeladas por parâmetros probabilísticos, que permitam a realização da análise discursiva. Em particular, eles são treinados e testados somente para a relação de discurso causa-efeito, similar à relação *cause-effect* de Kehler (2002). Além disso, os modelos são treinados para a língua inglesa, dada a necessidade de um grande conjunto de dados e de ferramentas de PLN de precisão suficiente.

Inicialmente, nas duas seções seguintes, o modelo *Noisy-Channel* e o método EM, nos quais os modelos estatísticos se baseiam, são introduzidos. A seguir, os modelos de análise discursiva são descritos.

5.1. Modelo Noisy-Channel

O modelo *Noisy-Channel* foi proposto por Shannon (1948) para a modelagem da capacidade de transmissão de dados em um canal. Originalmente, esse modelo foi aplicado para os sistemas de telefonia para tentar prever e corrigir os erros ocorridos durante a transmissão de mensagens. Na Figura 5.1, mostram-se as componentes do modelo. Inicialmente, tem-se uma mensagem M_1 produzida por uma fonte (*source*) com probabilidade $P(M_1)$; essa mensagem, ao ser transmitida por um canal com ruído (*noisy-channel*), é corrompida e erros são introduzidos, transformando-a em M_2 com probabilidade $P(M_2|M_1)$.

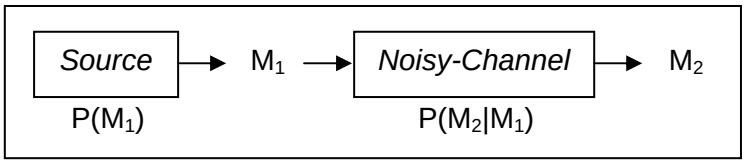


Figura 5.1 – Modelo *Noisy-Channel*

No caso da telefonia, a fonte normalmente é uma pessoa e o canal com ruído é a linha telefônica. Segundo esse modelo, sabendo-se as probabilidades P(M1) e P(M2|M1), é possível determinar M1 a partir de M2 por um processo conhecido como decodificação. Esse processo consiste em escolher a mensagem M1 que maximize as probabilidades P(M1) e P(M2|M1), ou seja, é um processo de busca em um espaço de soluções (Russel e Norvig, 2003).

O modelo *Noisy-Channel* foi aplicado com sucesso na área de processamento de fala (Jurafsky e Martin, 2000) e, mais recentemente, começou a ser aplicado em tarefas de processamento de língua natural escrita. Em especial, esse modelo causou grandes avanços na área de tradução automática estatística (Brown et al., 1990, 1993; Koehn et al., 2003), produzindo os tradutores automáticos com melhores resultados na área atualmente, segundo as avaliações internacionais realizadas pelo NIST[5] (*National Institute of Standards and Technology*). Nesta área, a tradução de uma sentença em francês para o inglês, por exemplo, é modelada da seguinte forma: uma sentença em inglês E é produzida por uma pessoa (*source*) com probabilidade P(E) e, ao ser comunicada (por meio de um canal com ruído hipotético), transforma-se em uma sentença em francês F com probabilidade P(F|E). Fazer a tradução de F para E consiste, portanto, em um processo de decodificação, considerando-se que P(E) e P(F|E) são conhecidos ou podem ser aprendidos. Em geral, as probabilidades P(E) e P(F|E) são estimadas pela aplicação do método (EM), introduzido na próxima seção.

Em PLN, o conjunto de probabilidades P(E) é chamado modelo lingüístico, pois informa a probabilidade de E ocorrer em uma língua; o conjunto de probabilidades P(F|E), por sua vez, é chamado modelo de tradução, pois indica como E se traduz/transforma em F. O processo de transformação de E em F é a parte principal do modelo *Noisy-Channel* e é chamado modelo ou história gerativa. No geral, pode-se afirmar que o sucesso do modelo *Noisy-Channel* na tarefa em que é aplicado depende da história gerativa que se assume para o problema tratado. Como

[5] http://www.nist.gov/

ilustração de uma história gerativa, considere a história comumente adotada por modelos simples de tradução automática de inglês para português para a sentença em inglês *Mary did not slap the green witch*:

(1) Inicialmente, algumas palavras são replicadas um número determinado de vezes ou eliminadas (a palavra *slap* foi replicada duas vezes e a palavra *did* foi eliminada)

Mary not slap slap slap the green witch

(2) A seguir, cada palavra em inglês é substituída por uma palavra em português (na seqüência, *Mary* por "Maria", *not* por "não", primeira ocorrência de *slap* por "deu", segunda ocorrência de *slap* por "um", terceira ocorrência de *slap* por "tapa", *the* por "na", *green* por "verde", *witch* por "bruxa")

Maria não deu um tapa na verde bruxa

(3) Por fim, as palavras são reordenadas, produzindo a sentença em português

Maria não deu um tapa na bruxa verde

As decisões de replicar e eliminar palavras, substituir palavras em inglês por suas correspondentes em português e reordenar as palavras da sentença são feitas com base nas probabilidades que compõem o modelo de tradução, isto é, P(F|E). Essas probabilidades são chamadas parâmetros do modelo.

O modelo *Noisy-Channel* também foi aplicado a outras tarefas de PLN, como perguntas e respostas (Soricut e Brill, 2004) e sumarização de textos (Daumé e Marcu, 2004). Em perguntas e respostas, explica-se como a resposta pode ser produzida a partir da pergunta; em sumarização, mostra-se como um sumário se relaciona ao texto ao qual se refere.

Para mais detalhes sobre o modelo *Noisy-Channel*, sugere-se a leitura da obra de referência e de Manning e Schütze (1999). Para mais detalhes sobre a aplicação do modelo para problemas de PLN, sugere-se a leitura de Marcu e Popescu (2005).

5.2. Método Expectation-Maximization (EM)

O método EM (*Expectation-Maximization*) (Dempster et al., 1977) é utilizado para estimar parâmetros de modelos probabilísticos em que há variáveis não observadas. Se todos os dados/variáveis de um problema fossem observados, seria simples estimar os parâmetros de seu modelo probabilístico subjacente (por cálculo de freqüências, por exemplo). Quando algum dado/variável não está disponível, o método EM costuma ser aplicado.

Em tradução automática estatística, como ilustrado anteriormente, assume-se que uma sentença é tradução de outra se e somente se é possível alinhar suas palavras, isto é, determinar as palavras da sentença em uma língua alvo (português, por exemplo) que correspondem às palavras da sentença em uma língua fonte (inglês, por exemplo). Nesse caso, o alinhamento entre as sentenças é a variável não observada. Note que há diversos alinhamentos possíveis entre duas sentenças, pois se devem considerar todas as possibilidades de correspondência entre as palavras (por exemplo, no exemplo anterior, *Mary* com "Maria", *Mary* com "não", *Mary* com "deu", ... , *witch* com "Maria", *witch* com "não", etc.). Se esses parâmetros de correspondência entre as palavras fossem conhecidos, seria possível determinar o melhor alinhamento entre duas sentenças; por outro lado, se o alinhamento entre duas sentenças fosse conhecido, seria possível estimar os parâmetros do modelo. Em casos como esse, em que não se tem nenhuma destas informações devido à presença da variável não observada, utiliza-se o método EM.

O método EM funciona da seguinte forma:

1. são atribuídas probabilidades iniciais para todos os parâmetros do modelo (assume-se, normalmente, a distribuição uniforme, em que todos os parâmetros têm igual probabilidade);
2. determinam-se as probabilidades de todos os alinhamentos possíveis para cada par de sentenças inglês-português do corpus de treinamento, em que a probabilidade de um alinhamento qualquer é dada pela multiplicação de todos os parâmetros que compõem o alinhamento (por exemplo, para o par de sentenças da subseção anterior, considerando-se somente a correspondência entre as palavras, a probabilidade do alinhamento considerado é dada por t(Maria|*Mary*) x t(não|*not*) x t(deu|*slap*) x t(um|*slap*) x t(tapa|*slap*) x t(na|*the*) x t(verde|*green*) x t(bruxa|*witch*),

em que t é o parâmetro que indica a probabilidade de tradução de uma palavra em outra);

3. com base nas probabilidades dos alinhamentos, estimam-se novas probabilidades para os parâmetros;
4. com base nos novos parâmetros, estimam-se novas probabilidades para os alinhamentos;
5. e assim por diante.

O cálculo das probabilidades dos alinhamentos é o passo *expectation* do método EM; a estimativa dos parâmetros com base nos alinhamentos é o passo *maximization* do método. É garantido que, a cada iteração do método EM, as probabilidades dos parâmetros se aproximam do valor ideal. Encerra-se o método quando as probabilidades convergem e estabilizam.

A base do aprendizado do método EM consiste na repetição de padrões no corpus de treinamento. A cada padrão que se repete, o método tem mais evidências para incrementar a probabilidade de determinados parâmetros do modelo, penalizando os parâmetros que representam eventos improváveis ou inadequados. Por exemplo, em todas as sentenças do corpus de treinamento, é provável que existam muitos alinhamentos entre as palavras *witch* e "bruxa"; para cada padrão deste encontrado, a probabilidade do parâmetro t(bruxa|*witch*) é incrementada, enquanto as probabilidades dos outros parâmetros possíveis t(Maria|*witch*), t(deu|*witch*), ... , t(verde|*witch*) não são, sendo estes parâmetros, portanto, penalizados.

Um dos problemas do método EM é que ele é de complexidade exponencial e, diante da quantidade de dados necessária para seu treinamento (principalmente em PLN), sua aplicação torna-se, na maior parte dos casos, inviável. Há diversas formas de se lidar com esse problema. A maioria das pesquisas na área reduz o número de valores possíveis para as variáveis não observadas e a quantidade de dados para treinamento. Por exemplo, no caso da tradução automática, costuma-se utilizar somente os alinhamentos mais prováveis entre duas sentenças (em vez de todos) e limitar o tamanho das sentenças para um número determinado de palavras.

Para mais detalhes sobre o método EM, sugere-se a leitura da obra de referência ou Manning e Schütze (1999). Na próxima seção, os modelos estatísticos desenvolvidos para análise discursiva são descritos.

5.3. Modelos de Análise Discursiva

Foram desenvolvidos três modelos estatísticos de análise discursiva segundo o modelo *Noisy-Channel*. Todos os modelos foram desenvolvidos, treinados e testados com base em um corpus de relações de discurso de causa-efeito para a língua inglesa. Apesar disso, os modelos são genéricos e independentes de língua, e, portanto, podem ser aplicados a qualquer relação discursiva ou língua. A escolha da relação causa-efeito para treino e teste dos modelos se deu devido à importância desta relação e ao fato de ela ser comum a todas as teorias discursivas. Por ser uma relação abrangente, ela é similar à relação semântica *cause-effect* proposta por Kehler (2002) (veja Seção 2.4 do Capítulo 2). Segundo Kehler, esse relação abrangeria um grande número de relações retóricas.

O modelo *Noisy-Channel* no qual os modelos de análise discursiva se baseiam é mostrado na Figura 5.2. Inicialmente, um evento de causa C é observado (produzido) com probabilidade P(C); esse evento é então corrompido e transforma-se no efeito E com probabilidade P(E|C). Nos três modelos de análise discursiva propostos, considera-se que a distribuição de P(C) é uniforme, isto é, todas os possíveis eventos C possuem a mesma probabilidade de serem observados. Com isso, P(E|C) é a principal componente do modelo, responsável por explicar como a causa C se transforma no efeito E. É nesse ponto em que os três modelos propostos divergem, ou seja, em como se supõe que um evento se transforma no outro.

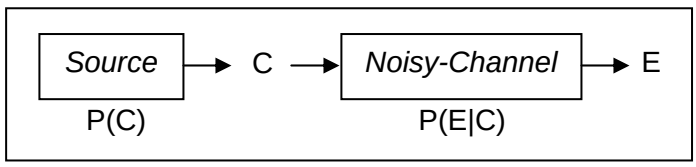


Figura 5.2 – Modelo *Noisy-Channel* para análise discursiva

Uma modificação que foi feita no modelo *Noisy-Channel* ao adaptá-lo para o problema da análise discursiva foi considerar uma probabilidade conjunta P(C,E) em vez da probabilidade condicional P(E|C). Isso foi feito porque, na probabilidade conjunta, não se assume como conhecida a dependência que existe entre os eventos, isto é, não se afirma que E é dependente de C (como ocorre na probabilidade condicional). Em relações causa-efeito, é evidente que o efeito se deve à ocorrência da

causa e, portanto, que o efeito é condicionado à causa. Entretanto, não se pode afirmar isso para outras relações do discurso, nas quais a direção da dependência não é clara ou não existe, como nas relações retóricas CONTRAST e SEQUENCE, por exemplo. Usar a probabilidade conjunta torna o modelo flexível o bastante para ser aplicado para qualquer uma das relações consideradas.

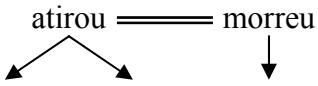
Nas subseções seguintes, os modelos desenvolvidos são detalhados.

5.3.1. Um Modelo Baseado em Palavras

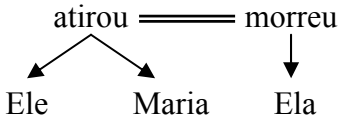
O primeiro modelo desenvolvido é baseado no relacionamento entre as palavras dos eventos. Sua história gerativa é delineada abaixo, exemplificada para as sentenças "Ele atirou em Maria." e "Ela morreu.", entre as quais há uma relação causa-efeito. Em cada passo da história gerativa, a produção da estrutura de causa-efeito subjacente às sentenças é exibida.

1) produz-se a relação causa-efeito entre "atirou" e "morreu" com probabilidade ce(atirou,morreu)

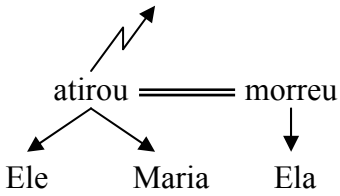
2) determina-se que "atirou" possui 2 argumentos (quem atirou e quem foi baleado) com probabilidade narg(2|atirou) e que "morreu" possui 1 argumento (quem morreu) com probabilidade narg(1|morreu)



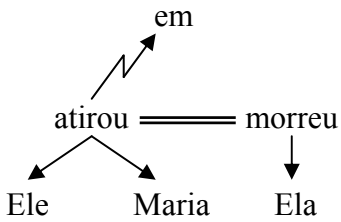
3) escolhem-se os argumentos dos eventos: “atirou” requer os argumentos “Ele” e “Maria” com probabilidades arg(Ele|atirou) e arg(Maria|atirou), respectivamente; “morreu” requer o argumento “Ela” com probabilidade arg(Ela|morreu)



4) determina-se que “atirou” produz uma palavra extra (isto é, que não é argumento) com probabilidade phi(1|atirou) e que “morreu” não produz palavras extras com probabilidade phi(0|morreu)



5) a palavra extra “em” é produzida com probabilidade ew(em)



Portanto, segundo essa história gerativa, a probabilidade P(C,E) é calculada pela seguinte fórmula:

$$
\begin{aligned}
P(C,E) = \quad & ce(causa,efeito) \times \\
& narg(M|causa) \times narg(N|efeito) \times \\
& \prod_{i=1}^{M} arg(w_i|causa) \times \prod_{i=1}^{N} arg(w_i|efeito) \times \\
& phi(|C|-M|causa) \times phi(|E|-N|efeito) \times \\
& \prod_{i=1}^{|C|-M} ew(w_i) \times \prod_{i=1}^{|E|-N} ew(w_i)
\end{aligned}
$$

em que M e N indicam o número de argumentos da causa e do efeito, respectivamente; |C| e |E| são o tamanho (número de palavras) da causa C e do efeito E, respectivamente; a letra w representa uma palavra.

Como ilustração, a probabilidade das sentenças "Ele atirou em Maria." e "Ela morreu." estarem relacionadas por uma relação causa-efeito, assumindo-se que a estrutura de causa-efeito subjacente é a mostrada na história gerativa, é dada pela fórmula abaixo:

$$
\begin{aligned}
P(C,E) = \quad & ce(atirou,morreu) \times \\
& narg(2|atirou) \times narg(1|morreu) \times \\
& arg(Ele|atirou) \times arg(Maria|atirou) \times arg(Ela|morreu) \times \\
& phi(1|atirou) \times phi(0|morreu) \times \\
& ew(em)
\end{aligned}
$$

As probabilidades ce, narg, arg, phi e ew são os parâmetros do modelo e são estimadas pela aplicação do método EM a partir de um corpus de sentenças relacionadas por relações causa-efeito (veja Seção 5.4 para descrição do corpus). Para aplicação do método EM, nos modelos de análise discursiva propostos, as estruturas de causa-efeito são as variáveis não observadas. Idealmente, espera-se que o método EM aprenda os parâmetros que representem mais adequadamente as estruturas de causa-efeito subjacentes às sentenças. Note que, para cada sentença, há diversas estruturas de causa-efeito possíveis (considerando diferentes eventos de causa e efeito, diferentes números de argumentos e diferentes argumentos, diferentes números de palavras extras e diferentes palavras extras). Para as sentenças "Ele atirou em Maria" e "Ela morreu", espera-se que o método aprenda, por exemplo, que: a probabilidade ce(atirou,morreu) seja maior do que ce(Ele,morreu), ce(Maria,morreu), ce(Ele,Ela) e ce(Maria,Ela); a probabilidade narg(2|atirou) seja maior do que narg(3|atirou), narg(1|atirou) e narg(0|atirou); a probabilidade arg(Ele|atirou) seja maior do que arg(em|atirou); etc. Os parâmetros aprendidos codificam o conhecimento semântico que se deseja aprender para a realização da análise discursiva automática.

Sabendo-se o valor dos parâmetros e, portanto, como calcular P(C,E), é possível identificar sentenças relacionadas por relações causa-efeito: se P(C,E) das sentenças é alta (isto é, há pelo menos uma estrutura de causa-efeito provável

subjacente às sentenças), então há uma relação causa-efeito entre as sentenças; caso contrário, não há uma relação causa-efeito.

Para tornar a história gerativa proposta mais informada e o método EM mais eficiente, algumas restrições foram impostas, a saber:

1) somente substantivos e verbos podem ser eventos de causa e efeito: para determinar as classes gramaticais das palavras, utiliza-se o tagger MXPOST, que segue o modelo de Ratnaparkhi (1996), com precisão de 97%;

2) somente substantivos (e pronomes), verbos, adjetivos e advérbios, ou seja, as palavras de classe aberta, podem ser argumentos de eventos de causa e efeito, e, assim, as palavras de classe fechada são sempre consideradas palavras extras do modelo.

5.3.2. Um Modelo Baseado em Conceitos

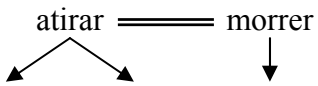
O segundo modelo proposto não se baseia apenas em palavras, mas nos conceitos correspondentes às palavras também. Com isso, espera-se que, em vez de se aprender que os argumentos de "atirou" é "Ele" e "Maria", aprenda-se que "atirou" tem argumentos do tipo "pessoa". Isso tornaria o modelo mais genérico e o conhecimento extraído mais intuitivo.

A história gerativa é ilustrada abaixo. Da mesma forma, é acompanhada de um exemplo e da estrutura de causa-efeito sendo construída. Note que, diferentemente da história gerativa anterior, manipula-se, agora, conceitos que, ao final, serão mapeados nas palavras das sentenças (conforme o novo parâmetro t).

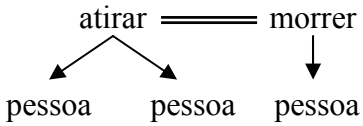
1) produz-se a relação causa-efeito entre os conceitos "atirar" e "morrer" com probabilidade ce(atirar,morrer)

atirar ═══ morrer

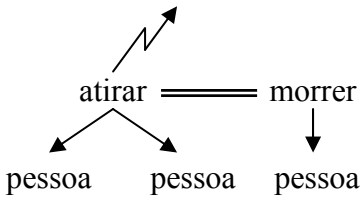
2) determina-se que “atirar” possui 2 argumentos com probabilidade narg(2|atirar) e que “morrer” possui 1 argumento com probabilidade narg(1|morrer)



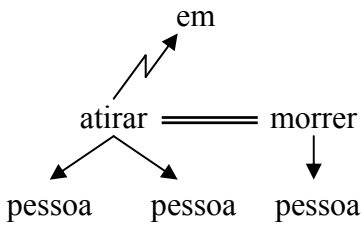
3) escolhem-se os argumentos dos eventos: “atirar” requer argumentos “pessoa” com probabilidades arg(pessoa|atirar); “morrer” também requer o argumento “pessoa” com probabilidade arg(pessoa|morrer)



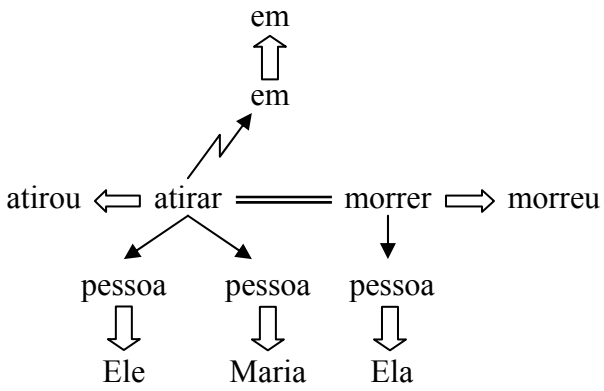
4) determina-se que “atirar” produz uma palavra extra (isto é, que não é argumento) com probabilidade phi(1|atirar) e que “morrer” não produz palavras extras com probabilidade phi(0|morrer)



5) o conceito extra “em” é produzido com probabilidade ew(em)



6) os conceitos são mapeados nas respectivas palavras com probabilidades t(atirou|atirar), t(morreu|morrer), t(Ele|pessoa), t(Maria|pessoa), t(Ela|pessoa) e t(em|em).



Segundo essa história gerativa, a probabilidade P(C,E) é calculada pela seguinte fórmula:

$$
\begin{aligned}
P(C,E) = \quad & ce(causa,efeito) \times t(w_{causa}|causa) \times t(w_{efeito}|efeito) \times \\
& narg(M|causa) \times narg(N|efeito) \times \\
& \prod_{i=1}^{M} (arg(c_i|causa) \times t(w_i|c_i)) \times \prod_{i=1}^{N} (arg(c_i|efeito) \times t(w_i|c_i)) \times \\
& phi(|C|-M|causa) \times phi(|E|-N|efeito) \times \\
& \prod_{i=1}^{|C|-M} (ew(c_i) \times t(w_i|c_i)) \times \prod_{i=1}^{|E|-N} (ew(c_i) \times t(w_i|c_i))
\end{aligned}
$$

Durante o treinamento do modelo, para se obter os conceitos correspondentes às palavras, utilizou-se a WordNet (Fellbaum, 1998). Para cada palavra, os três hiperônimos mais freqüentes foram buscados automaticamente.

Como ilustração, a probabilidade das sentenças “Ele atirou em Maria.” e “Ela morreu.” estarem relacionadas por uma relação causa-efeito, assumindo-se que a estrutura de causa-efeito subjacente é a mostrada na história gerativa, é dada pela fórmula a seguir:

P(C,E) = ce(atirar,morrer) x t(atirou|atirar) x t(morreu|morrer) x
narg(2|atirar) x narg(1|morrer) x
arg(pessoa|atirar) x t(Ele|pessoa) x arg(pessoa|atirar) x
t(Maria|pessoa) x arg(pessoa|morrer) x t(Ela|pessoa) x
phi(1|atirar) x phi(0|morrer) x
ew(em) x t(em|em)

De forma similar ao modelo anterior, assume-se que somente substantivos e verbos podem ser eventos de causa e efeito e que somente palavras de classe aberta (incluindo os pronomes) podem ser argumentos destes eventos. Os parâmetros do modelo também são aprendidos por meio do método EM. Todas as combinações entre conceitos e palavras são consideradas no aprendizado dos parâmetros.

5.3.3. Um Modelo Baseado na Estrutura Argumental dos Verbos

Os dois modelos anteriores fazem a suposição simplista de que a relação causa-efeito se estabelece entre palavras/conceitos das sentenças. Por exemplo, segundo esses modelos, o evento de atirar causa o evento de morrer, ou seja, ce(atirar,morrer).

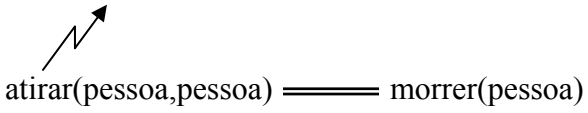
O novo modelo proposto considera que a relação causa-efeito se estabelece entre o fato de uma pessoa atirar em outra pessoa e o fato dessa pessoa morrer, isto é, ce(atirar($pessoa_1$,$pessoa_2$),morrer($pessoa_2$)), em que atirar($pessoa_1$,$pessoa_2$) e morrer($pessoa_2$) são as estruturas argumentais dos verbos atirar e morrer, respectivamente. A estrutura argumental de um verbo indica quantos são e quais são os possíveis argumentos que o verbo exige. O conceito de estrutura argumental será discutido com mais detalhes na Seção 5.6

A nova história gerativa é delineada a seguir.

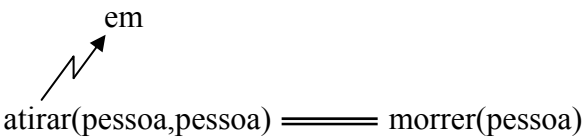
1) produz-se a relação causa-efeito entre o fato de uma pessoa atirar em outra pessoa e o fato dessa pessoa morrer com probabilidade ce(atirar(pessoa,pessoa),morrer(pessoa))

atirar(pessoa,pessoa) ═══ morrer(pessoa)

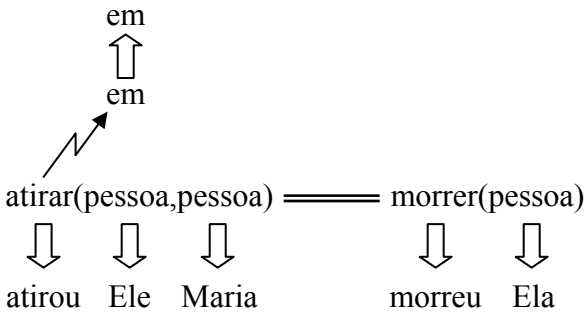
2) determina-se que “atirar” produz uma palavra extra (isto é, que não é argumento) com probabilidade phi(1|atirar) e que “morrer” não produz palavras extras com probabilidade phi(0|morrer)



3) o conceito extra “em” é produzido com probabilidade ew(em)



4) os conceitos são mapeados nas respectivas palavras com probabilidades t(atirou|atirar), t(morreu|morrer), t(Ele|pessoa), t(Maria|pessoa), t(Ela|pessoa) e t(em|em).



Para essa história gerativa, a probabilidade P(C,E) é calculada pela seguinte fórmula:

$$P(C,E) = ce(v_{causa}(Args_{causa}),v_{efeito}(Args_{efeito})) \times t(w_{causa}|v_{causa}) \times t(w_{efeito}|v_{efeito}) \times \prod_{i=1}^{\text{número de args da causa}} t(w_i|arg_i) \times \prod_{i=1}^{\text{número de args do efeito}} t(w_i|arg_i) \times phi(M|causa) \times phi(N|efeito) \times \prod_{i=1}^{M} (ew(c_i) \times t(w_i|c_i)) \times \prod_{i=1}^{N} (ew(c_i) \times t(w_i|c_i))$$

Como ilustração, a probabilidade das sentenças “Ele atirou em Maria.” e “Ela morreu.” estarem relacionadas por uma relação causa-efeito, assumindo-se que a estrutura de causa-efeito subjacente é a mostrada na história gerativa, é dada pela fórmula abaixo:

$$
\begin{aligned}
P(C,E) = \quad & ce(atirar(pessoa,pessoa),morrer(pessoa)) \times \\
& t(atirou|atirar) \times t(morreu|morrer) \times \\
& t(Ele|pessoa) \times t(Maria|pessoa) \times t(Ela|pessoa) \times \\
& phi(1|atirar) \times phi(0|morrer) \times \\
& ew(em) \times t(em|em)
\end{aligned}
$$

Para se determinar as possíveis estruturas argumentais em uma sentença, um parser estatístico foi usado. O parser segue o modelo de Collins (1999) e tem precisão de aproximadamente 91%, sendo o parser de maior precisão para a língua inglesa. Com o uso do parser, o modelo é capaz de lidar com as diferenças entre argumentos e adjuntos dos verbos (argumentos são obrigatórios, enquanto adjuntos não) e de identificar as orações relativas, que, em princípio, não devem ser consideradas na produção da estrutura argumental de uma sentença.

Considerou-se que somente substantivos e verbos podem ser predicadores em uma sentença e que todas as palavras de classe aberta (incluindo pronomes) podem ser argumentos em uma estrutura argumental. Entretanto, restringiu-se o número de argumentos de uma estrutura argumental para, no máximo, 3. Essa restrição foi sintaticamente motivada, pois, na maior parte das teorias sintáticas, um verbo não possui mais do que 3 argumentos.

Neste modelo, o método EM deve aprender tanto as melhores estruturas argumentais para as sentenças relacionadas (por exemplo, que a melhor estrutura argumental para a sentença “Ele atirou em Maria” é atirou(Ele,Maria)) quanto os melhores pares de causa-efeito (por exemplo, ce(atirou(Ele,Maria),morreu(Ela))).

5.4. Corpus

Para compor um corpus de relações causa-efeito, foram selecionadas automaticamente 400.000 sentenças em inglês de diversos corpora jornalísticos (*Reuters*, *New York Times*, *Wall Street Journal* e coleção TREC'2002 – *Text REtrieval Conference* (Voorhees e Buckland, 2002), entre outros) que contivessem a palavra *because* em seu interior, totalizando 11 milhões de palavras. Optou-se por selecionar as sentenças com a palavra *because* porque essa palavra é um dos marcadores discursivos mais fortes que sinalizam a relação causa-efeito na língua inglesa.

As sentenças coletadas foram anotadas com as ferramentas necessárias (tagger, parser e WordNet) para a execução de cada um dos modelos propostos e, a seguir, segmentadas em partes que correspondessem à causa (o trecho da sentença que está depois da palavra *because*) e ao efeito (o trecho de texto que está antes da palavra *because*) dentro da sentença. Por exemplo, na sentença *They prefer women commanders because they are less strict*, o seguinte par de causa-efeito é produzido:

they are less strict – they prefer women commanders

Foram selecionadas sentenças com, no máximo, 35 palavras. Essa limitação é necessária devido ao método EM, que é de complexidade exponencial, como já discutido.

De forma similar, foram coletadas 900.000 sentenças com a palavra *but* em seu interior, para formar um corpus de relações de oposição. Parte desse corpus foi usada para testar os modelos treinados com os pares de causa-efeito. Esses modelos devem (a) reconhecer os pares de causa-efeito como sendo realmente de causa-efeito e (b) rejeitar os pares de oposição. A seguir, a avaliação dos modelos é apresentada.

5.5. Avaliação dos Modelos de Análise Discursiva

Os três modelos propostos foram treinados com o corpus de relações causa-efeito e testados com esse mesmo corpus e com o corpus de relações de oposição, produzindo os resultados mostrados na Tabela 5.1. O modelo baseado em palavras obteve uma

taxa de acerto de 59% ao ser aplicado ao corpus de teste de relações causa-efeito (isto é, em 59% dos casos, o modelo detectou a relação causa-efeito entre as sentenças) e uma taxa de acerto de 61% ao ser aplicado ao corpus de teste de relações de oposição (isto é, em 61% dos casos, o modelo detectou que aquelas relações não eram de causa-efeito); o modelo baseado em conceitos obteve taxas de acerto de 71% e 43% ao ser aplicado aos corpora de relações causa-efeito e de oposição, respectivamente; e o modelo baseado nas estruturas argumentais obteve taxas de acerto de 61% e 50% ao ser aplicado aos corpora de relações causa-efeito e de oposição, respectivamente.

Para a realização desta avaliação, determinou-se que duas sentenças estariam relacionadas por uma relação causa-efeito se a probabilidade P(C,E) fosse maior do que um *threshold* (isto é, um valo mínimo), o qual foi calculado empiricamente a partir um corpus de relações causa-efeito (disjunto do corpus de treino) de 100 sentenças. De fato, em vez de se usar a probabilidade P(C,E), utilizou-se a medida de perplexidade (Manning e Schütze, 1999), que é, basicamente, a probabilidade P(C,E) normalizada em relação ao tamanho das sentenças relacionadas. Diferentemente da probabilidade P(C,E), a perplexidade é independente do tamanho das sentenças, tornando a avaliação mais justa. Para mais detalhes sobre esta medida, sugere-se a leitura da obra referenciada.

Tabela 5.1 – Taxa de acerto dos modelos de análise de discursiva

	causa-efeito	**oposição**
Modelo baseado em palavras	59%	61%
Modelo baseado em conceitos	71%	43%
Modelo baseado em estruturas argumentais	61%	50%

Algumas observações feitas a partir destes resultados são:

- há um vocabulário comum entre as palavras das relações causa-efeito e de oposição, o que não permite que o modelo baseado em palavras tenha uma taxa de acerto maior;
- a utilização de conceitos melhorou a identificação das relações causa-efeito, mas piorou o desempenho do modelo na identificação de relações que não são de causa-efeito;
- no modelo baseado nas estruturas argumentais, o método EM não foi capaz de aprender boas estruturas argumentais e bons pares de causa-efeito ao mesmo tempo.

Os dois primeiros modelos não tiveram uma taxa de acerto maior devido, principalmente, às suposições simplistas que fazem, como já foi discutido anteriormente. Em relação ao terceiro modelo, por sua vez, o problema reside no fato dos dados serem esparsos para que o aprendizado seja eficaz. Dados esparsos apresentam pouca redundância, que é a condição necessária para que técnicas de Aprendizado de Máquina tenham um bom desempenho.

Em relação ao terceiro modelo, verificou-se que a forma como a análise discursiva é realizada pode ser dividida em dois subproblemas: (I) determinar as possíveis estruturas argumentais e (II) determinar os pares de causa-efeito. Caso as estruturas argumentais possam ser previamente aprendidas, restaria ao modelo baseado em estruturas argumentais aprender os pares de causa-efeito somente. Acredita-se que essa simplificação possa melhorar significativamente o desempenho desse modelo, já que os dados se tornam menos esparsos. Diante disto, um modelo para o aprendizado das estruturas argumentais foi proposto. Este modelo é apresentado a seguir. A utilização das estruturas argumentais aprendidas para a análise discursiva como ilustrada anteriormente consiste em um dos trabalhos futuros decorrentes desta tese de doutorado.

5.6. Um Modelo para o Aprendizado das Estruturas Argumentais dos Verbos

Foi proposto um modelo estatístico baseado no modelo *Noisy-Channel* para resolver o problema de se aprender possíveis estruturas argumentais. Considerou-se, nesse caso, somente a classe dos verbos como predicadora (isto é, as palavras que exigem argumentos).

Conhecendo-se as estruturas argumentais mais prováveis das sentenças, basta que, no modelo de análise discursiva baseado em estruturas argumentais, o método EM aprenda apenas quais os melhores pares de causa-efeito. Com isso, o problema dos dados serem esparsos tem um impacto significativamente menor.

Há diversas propostas na literatura para a aquisição das estruturas argumentais dos verbos. Esses trabalhos foram estudados e avaliados para se determinar suas

vantagens e desvantagens para a tarefa em questão, assim como para se entender a problemática envolvida na tarefa. Os trabalhos mais relevantes são brevemente descritos na subseção seguinte.

5.6.1. Trabalhos correlatos

Há alguns projetos para o desenvolvimento em larga escala de repositórios de informação semântica de verbos em inglês, visando a codificar suas estruturas argumentais possíveis, suas estruturas de subcategorização e exemplos reais de uso dos verbos. Os projetos mais conhecidos são a FrameNet[6] (Baker et al., 1998), a VerbNet[7] (Kipper et al., 2000) e o PropBank[8] (Kingsbury e Palmer, 2002). Como exemplo, as Figuras 5.3-5.5 mostram partes das anotações da FrameNet, da VerbNet e do PropBank para o verbo *buy* ("comprar", em português), respectivamente. A FrameNet mostra o padrão no qual o verbo ocorre e exemplos; a VerbNet mostra os papéis temáticos dos argumentos que o verbo requer e seus traços semânticos, as possíveis estruturas de subcategorização e exemplos para cada uma; o PropBank exibe os papéis dos argumentos, as possíveis estruturas de subcategorização e exemplos para cada uma. Note que o PropBank também distingue os complementos obrigatórios (os argumentos, propriamente ditos) dos opcionais (os adjuntos). O complemento opcional na Figura 5.5 é o argumento identificado por ArgM-MNR (isto é, argumento de modo – do inglês, *manner*).

Typical pattern:

BUYER buys GOODS from SELLER for MONEY

Example:

Abby bought a car from Robin for $5,000.

Figura 5.3 – Anotação da FrameNet para o verbo *buy*

[6] http://framenet.icsi.berkeley.edu/
[7] http://www.cis.upenn.edu/group/verbnet/
[8] http://www.cis.upenn.edu/~ace/

Thematic Roles:
Agent[+animate OR +organization]
Asset[-location -region]
Beneficiary[+animate OR +organization]
Source[+concrete]
Theme[]

Frames:

Basic Transitive:

"Carmen bought a dress"
Agent V Theme

Benefactive Alternation (double object):

"Carmen bought Mary a dress"
Agent V Beneficiary Theme

Figura 5.4 – Anotação da VerbNet para o verbo *buy*

Roles:
Arg0:buyer
Arg1:thing bought
Arg2:seller
Arg3:price paid
Arg4:benefactive

Examples:

Intransitive:

Consumers who buy at this level are more educated than they were.

Arg0:	Consumers
REL:	buy
ArgM-MNR:	at this level

Basic transitive:

They bought $2.4 billion in Fannie Mae bonds

Arg0:	They
REL:	bought
Arg1:	$2.4 billion in Fannie Mae bonds

Figura 5.5 – Anotação do PropBank para o verbo *buy*

Estes repositórios têm sido construídos manualmente, dada a dificuldade e subjetividade da tarefa. Os problemas da abordagem de construção manual são que ela é custosa, pois precisa de humanos treinados, consome muito tempo e está sujeita à inserção de dados errados, inconsistentes ou incompletos. Além disso, ao se comparar

as informações dos repositórios, podem-se notar diferentes níveis de abstração na anotação (por exemplo, para descrever o objeto comprado – veja Figuras 5.3-5.5 – tem-se *goods* na FrameNet, *theme* na VerbNet e *thing bought* no PropBank) e diferentes esquemas de anotação (por exemplo, o PropBank distingue os argumentos dos adjuntos em suas estruturas, enquanto a FrameNet e a VerbNet não).

Alguns trabalhos propuseram formas automáticas (por exemplo, Brent, 1991; Resnik, 1992; Grishman e Sterling, 1992; Manning, 1993; Framis, 1994; Briscoe e Carroll, 1997; Rooth et al., 1999; McCarthy, 2000; Sarkar e Zeman, 2000; Merlo e Stevenson, 2001; Sarkar e Tripasai, 2002; Gildea, 2002) e semi-automáticas (por exemplo, Korhonen, 2002; Green et al., 2004; Gomez, 2004) para a derivação das estruturas argumentais dos verbos. No geral, estes trabalhos fazem uso de parsers e/ou dicionários de subcategorização (isto é, dicionários que especificam os comportamentos dos verbos, isto é, os argumentos que exigem e como estes se realizam sintaticamente em um sentença) para identificar os argumentos de um verbo em uma sentença, ou assumem como conhecidos os tipos da estrutura que um verbo pode possuir (em termos de número e ordem de argumentos). Alguns trabalhos (por exemplo, Grishman e Sterling, 1994; Framis, 1994; Lapata, 1999; Gomez, 2004) também tentam fazer generalizações nas estruturas aprendidas, calculando a similaridade entre as palavras de diferentes estruturas ou usando recursos lexicais como a WordNet. A maioria deles também possui algum passo de filtragem dos resultados, no qual algumas estruturas aprendidas são descartadas manual ou automaticamente (por meio de medidas baseadas em freqüência).

As desvantagens destas abordagens são claras: apenas algumas línguas possuem bons parsers, *wordnets* ou dicionários de subcategorização disponíveis; é necessário que se saibam os tipos de estrutura que um verbo possui; qualquer análise manual é custosa nesse cenário. A abordagem proposta neste trabalho, descrita a seguir, supera alguma destas dificuldades.

5.6.2. Um Modelo para Aprendizado Não Supervisionado de Estruturas Argumentais

Na Figura 5.6, o modelo proposto é esquematizado.

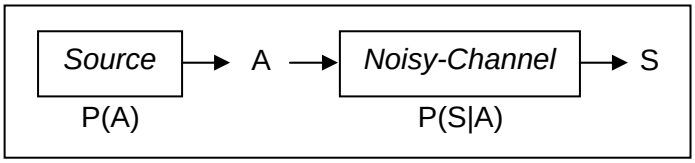


Figura 5.6 – Modelo *Noisy-Channel* para aprendizado das estruturas argumentais dos verbos

Nesse modelo, assume-se que uma estrutura argumental A qualquer é produzida por uma fonte com probabilidade P(A) e que, após ser corrompida em um canal com ruído, A se transforma na sentença S com probabilidade P(S|A). Nesse modelo, explica-se, portanto, como uma estrutura argumental é realizada superficialmente em uma sentença S. A história gerativa para isso é mostrada abaixo, exemplificada para a derivação da sentença "O menino comprou um brinquedo".

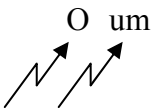
1) produz-se a estrutura argumental comprou(menino,brinquedo) com probabilidade arg(comprou(menino,brinquedo))

comprou(menino,brinquedo)

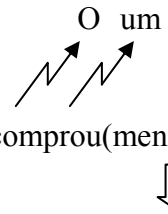
2) determina-se que "comprou" produzirá duas palavras extras (isto é, adjuntos) com probabilidade phi(2|comprou)

comprou(menino,brinquedo)

3) escolhem-se as palavras extras “O” e “um” com probabilidades ew(O) e ew(um), respectivamente



4) as palavras são reordenadas para produzir a sentença com distribuição uniforme



Portanto, os parâmetros deste modelo que devem ser aprendidos são três: arg, phi e ew. O que se objetiva conseguir, de fato, são os parâmetros arg mais prováveis para cada verbo da língua. Note que os parâmetros phi e ew são essenciais para o modelo, pois é por meio deles que o modelo é capaz de diferenciar os argumentos dos adjuntos. Entretanto, concluído o aprendizado, os parâmetros phi e ew são dispensáveis para a tarefa em foco, pois os parâmetros arg codificam toda a informação desejada.

De acordo com a história gerativa delineada, a probabilidade de uma sentença S é dada pela seguinte fórmula:

$$P(S) = \sum_{A} P(S,A) = \sum_{A} P(A) \times P(S|A)$$

$$= \sum_{A} \text{arg}(A) \times \text{phi}(N|\text{verbo}) \times \prod_{i=1}^{N} \text{ew}(w_i)$$

em que A é uma estrutura argumental, N é o número de palavras extras que são geradas e w_i é a palavra de número i sendo gerada. Desta forma, a probabilidade de

uma sentença existir é a somatória da probabilidade de todas as suas estruturas argumentais possíveis.

O modelo é treinado por meio do método EM, já explicado anteriormente. Neste contexto, todas as possíveis estruturas argumentais para uma sentença são consideradas. Na Figura 5.7, mostram-se todas as estruturas possíveis para a sentença "Ele comprou presentes para ela". As palavras apontadas pelas setas são os argumentos da estrutura argumental; as palavras não apontadas são as palavras extras.

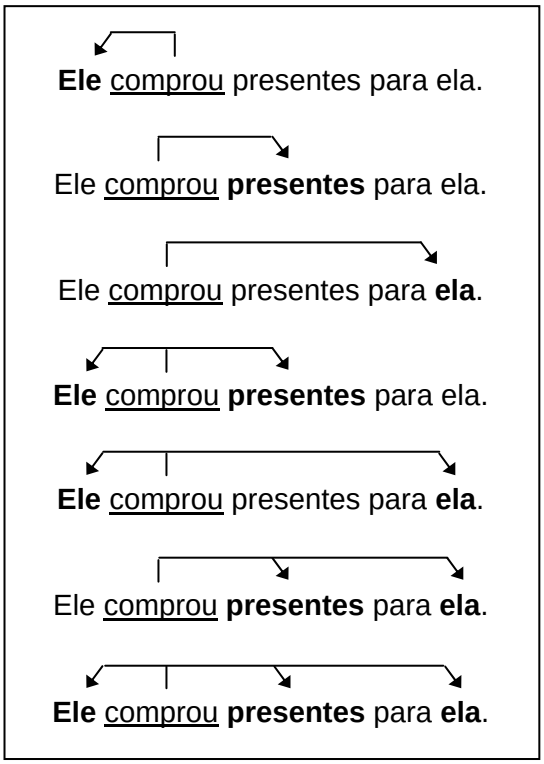


Figura 5.7 – Estruturas argumentais possíveis para a sentença "Ele comprou presentes para ela."

Para tornar o aprendizado mais eficiente, as seguintes restrições foram adotadas (como pode ser observado na figura): uma estrutura argumental pode ter, no máximo, 3 argumentos; os argumentos podem ser somente palavras de classe aberta, isto é, substantivos (incluindo pronomes), adjetivos, verbos e advérbios. Para se identificar as palavras de classe aberta, um tagger é utilizado, como explicado na subseção seguinte, a qual descreve o corpus de treinamento do modelo.

5.6.3. Corpus

Para formar o corpus de treinamento do modelo, foram coletadas todas as sentenças da coleção TREC'2002 para os 1.500 verbos mais freqüentes do inglês. Devido à complexidade exponencial do método EM, foram selecionadas sentenças de tamanho máximo de 10 palavras. O corpus resultante contém 14 milhões de palavras, com uma média de 1.400 sentenças por verbo.

Para identificação das palavras de classe aberta, as únicas que podem ser argumentos (segundo a restrição adotada no modelo proposto), as sentenças foram etiquetadas por um tagger, o MXPOST, que segue o modelo de Ratnaparkhi (1996). Além disso, para tornar o modelo apto a fazer generalizações, optou-se por dados anotados por um reconhecedor de entidades mencionadas (REM) (no inglês, *named-entity recognizer*), isto é, um sistema que identifica, em uma sentença, as palavras que representam entidades das classes "organização", "data", "lugar", "pessoa", etc. Foi por esta razão que as sentenças do corpus de treinamento foram coletadas da coleção TREC'2002, pois esta já havia sido anotada pelo REM *BBN IdentiFinder* (Bikel et al., 1999).

Outras modificações que foram feitas nos dados são:

- todos os numerais foram substituídos pela classe genérica *number*;
- com exceção dos pronomes *it*, *they* e *them*, os pronomes foram substituídos pela classe genérica *person*; os pronomes *it*, *they* e *them* foram considerados como podendo ser tanto da classe *person*, quanto da classe *thing* (isto é, qualquer coisa que não seja *person*), já que eles podem se referir a qualquer entidade.

Optou-se pelo uso de um REM, em oposição à WordNet, pelo fato do REM indicar univocamente as classes das palavras, não inserindo, portanto, ambigüidade no processamento. Na WordNet, para cada palavra, há diversas classes possíveis hierarquizadas.

Na Figura 5.8, exibe-se uma amostra dos dados de treinamento do modelo, com as etiquetas morfossintáticas após a barra e as entidades mencionadas em negrito.

about/IN ***money***/NN *home*/NN *water*/NN *heaters*/NNS *are*/VBP *bought*/VBN *each*/DT *year*/NN

person/PRP *bought*/VBD ***thing***/PRP ***number***/CD *years*/NNS *ago*/RB

organization/NNP *bought*/VBD ***organization***/NN *from*/IN ***organization***/NN last/JJ *year*/NN

thing/PRP *bought*/VBD *the*/DT *outstanding*/JJ *shares*/NNS *on*/IN ***date***/NNP

the/DT *cafeteria*/NN *bought*/VBD *extra*/JJ *plates*/NNS

Figura 5.8 – Amostra dos dados de treinamento do modelo de aprendizado de estruturas argumentais

Algumas coisas interessantes de se notar são: algumas sentenças são completamente lexicalizadas, sem entidades mencionadas (por exemplo, última sentença da figura); algumas sentenças possuem várias entidades mencionadas (por exemplo, segunda sentença da figura); na primeira sentença da figura, o termo *8 million* foi classificado erroneamente pelo REM como sendo da classe *money*. Esses erros não interferem no aprendizado, pois, por serem pouco freqüentes, são naturalmente descartados pelo método EM.

Como resultado deste treinamento, foi produzido um repositório de estruturas argumentais chamado ArgBank, contendo as estruturas argumentais aprendidas para os 1.500 verbos do inglês para os quais o modelo foi treinado. Este repositório poderá servir de entrada para um novo modelo estatístico de aprendizado de relações discursivas, como se discute no Capítulo 6.

Na próxima subseção, relata-se a avaliação do modelo proposto.

5.6.4. Avaliação e Discussão

Para avaliar o modelo, as medidas clássicas de precisão e cobertura foram calculadas para 20 verbos selecionados aleatoriamente, garantindo-se que verbos de freqüência baixa, média e alta do inglês estivessem incluídos no conjunto. Na primeira coluna da Tabela 5.2, mostram-se os verbos utilizados na avaliação. Os verbos *aspire*, *hook* e *yell* são exemplos de verbos de baixa freqüência; *raise* e *spend* são verbos de freqüência média; *offer*, *help* e *die* são verbos que ocorrem com freqüência alta.

Tabela 5.2 – Desempenho do modelo de aprendizado de estruturas argumentais

Verbos	Número de sentenças	Número de estruturas	Modelo estatístico		*Baseline*	
			Precisão (%)	Cobertura (%)	Precisão (%)	Cobertura (%)
aspire	25	3	100	100	100	100
hook	46	2	100	33,3	100	0
yell	110	5	100	50,0	100	100
spin	111	4	100	66,6	100	33,3
collapse	153	4	91,6	75,0	66,6	50,0
abandon	171	3	100	50,0	100	0
paint	253	5	93,3	33,3	100	16,6
fix	270	18	86,9	40,0	75,8	20,0
avoid	482	6	100	100	100	100
hate	594	27	91,3	100	71,5	100
issue	955	10	100	75,0	73,3	50,0
earn	971	43	88,3	75,0	76,7	50,0
cause	1.301	29	93,0	100	63,1	100
raise	1.422	63	93,0	83,3	66,6	50,0
spend	1.560	22	96,9	100	77,2	25,0
buy	2.326	44	85,5	70,0	75,6	70,0
expect	2.597	64	84,3	100	70,8	100
offer	3.071	41	95,8	20,0	77,9	20,0
help	3.706	54	89,4	100	76,4	100
die	4.334	70	85,2	100	57,5	100
Média	**1.223**	**26**	**93,7**	**73,5**	**81,4**	**59,2**

Precisão, neste caso, indica quantas das estruturas aprendidas pelo modelo são plausíveis; cobertura indica quantas das estruturas que deviam ser aprendidas foram aprendidas, de fato, pelo modelo. Para o cálculo da precisão, três especialistas humanos (lingüistas computacionais) julgaram a plausibilidade das estruturas aprendidas, podendo classificar cada estrutura como "plausível", "não plausível" ou "não sei dizer". Para o cálculo da cobertura, o repositório utilizado como referência foi o PropBank. Neste caso, procurou-se por estruturas aprendidas similares às estruturas previstas pelo PropBank, em termos de número, ordem e tipo semântico dos argumentos. Em ambas as medidas, consideraram-se somente as estruturas com probabilidade maior do que o *threshold* de 10^{-3} para que a avaliação fosse possível.

Para fins de comparação e validação do modelo proposto, utilizou-se um método *baseline* para produção das estruturas argumentais dos verbos. Esse método produz todas as estruturas argumentais nas quais as sentenças podem se basear, da mesma forma que se mostra na Figura 5.7, e ranqueia as estruturas de acordo com suas freqüências. As etiquetas morfossintáticas e as entidades mencionadas também

são consideradas na produção das estruturas argumentais, ou seja, o método *baseline* tem informação suficiente para decidir que palavras podem ser argumentos e pode produzir estruturas generalizadas. Como se verá pelos resultados da avaliação, este método se mostrou bastante robusto, mas não superou os resultados produzidos pelo modelo estatístico.

Os resultados destas avaliações são apresentados nas duas últimas colunas da Tabela 5.2. Na segunda e terceira colunas da tabela, mostram-se o número de sentenças de treinamento para o verbo em questão e o número de estruturas avaliado. Na média, o modelo atingiu precisão de 93,7% e cobertura de 73,5%. Para o julgamento dos 3 juízes, a medida de concordância Kappa (Carletta, 1996) foi calculada em 0,69. Um valor entre 0,60 e 0,80 indica uma boa concordância. O método *baseline* atingiu precisão de 81,4% e cobertura de 59,2%.

Calcularam-se, também, as medidas de precisão e cobertura para as 10 estruturas mais prováveis e para as 20 estruturas mais prováveis dos verbos para se verificar como a consideração de mais estruturas altera as medidas de precisão e cobertura. Na Tabela 5.3, mostram-se os resultados dessa avaliação. Como esperado, quanto mais estruturas são consideradas na avaliação, menor é a precisão (pois mais estruturas com menor probabilidade são consideradas) e maior é a cobertura.

Tabela 5.3 – Desempenho do modelo para as 10 e 20 estruturas mais prováveis

Número de estruturas	**Modelo estatístico**		***Baseline***	
	Precisão (%)	**Cobertura (%)**	**Precisão (%)**	**Cobertura (%)**
10	95,2	63,1	86,6	47,5
20	93,8	65,6	84,8	52,4
Todas	93,7	73,5	81,4	59,2

Na geral, os erros detectados nas estruturas argumentais durante o cálculo da precisão foram causados pelos seguintes problemas:

- presença de advérbios nas estruturas argumentais: idealmente, advérbios não deveriam estar presentes nas estruturas argumentais dos verbos, pois são considerados adjuntos; entretanto, em sentenças como *He asked rhetorically* e *He asked incredulously*, os advérbios *rhetorically* e *incredulously* parecem essenciais para o significado das sentenças e, além disso, ocorrem tão freqüentemente com esse verbo que foram considerados argumentos durante o aprendizado;

- ocorrência de *phrasal verbs* no corpus de treinamento (isto é, verbos que, associados a algumas partículas, adquirem sentidos diversos): o método de aprendizado não é capaz de distinguir esse tipo de fenômeno lingüístico dos verbos comuns, acarretando o aprendizado errôneo de estruturas como *gave*(*he*,*up*) e *gave*(*he*) para a sentença *He gave up*, por exemplo, sendo que ambas são consideradas estruturas inadequadas.

Em relação ao cálculo da cobertura, notou-se que, em alguns casos, as estruturas no PropBank contêm mais do que 3 argumentos, o que ocorre pela inclusão de adjuntos nas estruturas (vale lembrar que, no modelo proposto neste trabalho, consideram-se, no máximo, 3 argumentos). Por exemplo, para a sentença *John killed Mary with a pipe in the conservatory*, 4 argumentos (*John*, *Mary*, *pipe* e *conservatory*) são previstos no PropBank para o verbo *killed*.

Como ilustração, na Figura 5.9, mostram-se as 10 estruturas mais prováveis aprendidas pelo modelo proposto para o verbo *buy* e suas probabilidades.

1	*buy*(*organization*,*organization*)	1.20e-01
2	*buy*(*person*,*number*)	8.44e-02
3	*buy*(*person*,*thing*)	7.10e-02
4	*buy*(*organization*,*thing*)	5.63e-02
5	*buy*(*person*,*organization*)	4.28e-02
6	*buy*(*organization*,*person*)	3.51e-02
7	*buy*(*person*,*house*)	1.54e-02
8	*buy*(*person*,*thing*,*anyway*)	1.54e-02
9	*buy*(*money*,*money*)	1.40e-02
10	*buy*(*organization*,*organization*,*date*)	8.63e-03

Figura 5.9 – Estruturas argumentais mais prováveis aprendidas para o verbo *buy*

Algumas coisas interessantes de se notar nessas estruturas são:

- as estruturas 5 e 6 são similares: na primeira, uma pessoa compra uma organização (voz ativa); na segunda, uma organização é comprada por uma pessoa (voz passiva);
- a estrutura 7 possui um item lexicalizado (*house*);
- na estrutura 8, há uma inadequação causada pela inclusão de um advérbio como argumento (*anyway*);
- na estrutura 9, há um erro causado pelo *phrasal verb buy down* (como em *the dollar bought down the yen*).

Para muitos verbos, o modelo foi capaz de aprender sentidos não previstos no PropBank. Por exemplo, para o verbo *raise*, foram aprendidas estruturas para seu sentido de "crescer" (como em *Peter was raised in a big city*). Também se aprenderam diversas variações do uso dos verbos não listadas no PropBank. Por exemplo, para o verbo *die*, tem-se:

(a) In *date*, *person* died.
(b) *Person* died in *date*.
(c) *Person* died in *date* in *location*.
(d) *Person* died in *location* in *date*.

Além destas vantagens, o modelo proposto se destaca pelos seguintes pontos: (a) seu aprendizado é completamente automático, (b) por se basear em corpus, evidências lingüísticas são fornecidas para cada estrutura argumental aprendida, (c) o nível de abstração mais apropriado (itens lexicais vs. entidades) é determinado automaticamente de forma consistente, (d) não se faz necessário o uso de ferramentas sofisticadas de processamento de língua natural e (e) além das estruturas argumentais, têm-se probabilidades associadas a elas. Devido a (b), repositórios de informações semânticas para verbos (como FrameNet, VerbNet e PropBank) podem ser produzidos automaticamente para línguas que não os têm ou podem ser complementados para as línguas que já os tem. Por causa de (e), é possível utilizar o conhecimento aprendido em diversas aplicações, por exemplo: em tradução automática, as probabilidades das estruturas argumentais podem ser associadas às sentenças decodificadas, melhorando o resultado do processo; em parsers, podem-se selecionar as análises mais prováveis com base nas probabilidades das estruturas argumentais.

Em relação ao uso deste modelo para auxiliar o modelo de análise discursiva baseado em estruturas argumentais, é interessante notar que a inclusão de advérbios nas estruturas aprendidas pode ser importante para a análise pretendida. Por exemplo, a presença de advérbios como "nunca" e "não" em um segmento alteram seu significado, normalmente. Idealmente, na análise discursiva, isso deve ser considerado durante o aprendizado.

Apresentam-se, no próximo capítulo, as conclusões e considerações finais deste trabalho.

6. Conclusões

Foram apresentados, nesta tese de doutorado, métodos para a análise discursiva automática, uma área de pesquisa que tem recebido muita atenção ultimamente. Relatou-se o desenvolvimento do primeiro analisador retórico automático para o português do Brasil e propuseram-se modelos estatísticos para a análise discursiva automática.

Nas seções seguintes, discutem-se as potencialidades e limitações dos métodos discutidos e as contribuições dadas à área de pesquisa.

6.1. Sobre o DiZer

O DiZer pertence à abordagem simbólica da Inteligência Artificial, na qual se extrai e se formaliza o conhecimento necessário para a realização da tarefa em foco. Esse processo é, normalmente, custoso e, às vezes, manual, exigindo um engenheiro de conhecimento. Por outro lado, produz uma grande quantidade de conhecimento que pode ser reutilizável.

Para o desenvolvimento do DiZer, realizou-se uma análise manual de um corpus de 100 textos científicos da Computação anotado retoricamente, o CorpusTCC, representando-se o conhecimento discursivo em padrões de análise que possibilitaram a automação da tarefa visada.

O DiZer foi avaliado e demonstrou resultados satisfatórios tanto para textos científicos quanto para textos de outro gênero, no caso, textos jornalísticos. Os resultados apresentados validam a metodologia de desenvolvimento do DiZer, possibilitando o uso do sistema em outras aplicações de PLN.

6.1.1. Bases de Conhecimento

Com o desenvolvimento do DiZer, uma grande quantidade de conhecimento discursivo foi gerado.

Mapearam-se os marcadores textuais que sinalizam as relações retóricas, apresentando-se, principalmente, (a) a proporção de segmentos marcados, (b) a distribuição desses marcadores nos segmentos envolvidos e (c) como a nuclearidade das relações é expressa nos textos. Todo esse conhecimento foi codificado em mais de 740 padrões de análise.

O CorpusTCC anotado retoricamente pode subsidiar outras pesquisas em análise discursiva, teóricas ou aplicadas em problemas de PLN. Além disso, há o RHETALHO, um corpus de referência desenvolvido para a avaliação do DiZer, que pode ser igualmente explorado.

Foram estudados e implementadas soluções para o problema clássico da segmentação textual de forma a se delimitar as orações de um texto, as quais, em geral, expressam as proposições simples que são os constituintes básicos de uma estrutura retórica. Investigaram-se técnicas para detecção de relações retóricas entre proposições de um texto e formas de se construir suas estruturas retóricas possíveis.

6.1.2. Ferramental

Além do DiZer e do conhecimento produzido, foi desenvolvida uma ferramenta visual de auxílio à análise de corpus chamada RhetDB. Esta ferramenta oferece a possibilidade de importação dos dados produzidos pela *RST Annotation Tool* de Marcu para manipulação computacional e/ou análise lingüística dos dados.

Grande parte do conhecimento produzido neste trabalho de doutorado foi adquirida pelo uso desta ferramenta.

6.1.3. Aplicações em PLN

Devido ao bom desempenho do DiZer mostrado pela avaliação realizada, ele pode ser utilizado para outros fins, como sumarização automática de textos e resolução anafórica, entre outras aplicações.

Muitos trabalhos em sumarização automática têm mostrado que o conhecimento discursivo pode ser muito útil nesta área de pesquisa. Veja, por exemplo, Marcu (2000b) e Pardo (2002). A partir da estrutura retórica de um texto, os métodos apresentados nestes trabalhos selecionam as informações relevantes para a

produção do sumário correspondente. A possibilidade de uso do DiZer nesta aplicação é reforçada pelo fato de o DiZer se basear na teoria de discurso RST, para a qual as pesquisas têm convergido.

Cristea et al. (1998) propõem o tratamento computacional de anáforas em um texto pelo uso de sua estrutura retórica. Os autores mostram que a estruturação retórica de um texto permite que se determine o local em que o referente de uma anáfora pode estar. Baseando-se nisso, Seno e Rino (2005) propõem um método de sumarização que faz uso da teoria de Cristea et al. e do método de sumarização de Marcu para preservar a coerência de sumários produzidos automaticamente. O DiZer pode ser utilizado neste caso para fornecer a estrutura retórica de um texto necessária para que as teorias e métodos propostos sejam aplicados.

Outras possibilidades são, por exemplo, o uso do DiZer em:

- aplicações de tradução automática, modelando a estrutura organizacional do texto na língua fonte para posterior adaptação à estrutura do texto na língua destino (veja, por exemplo, Marcu et al., 2000);
- sistemas de auxílio à escrita, detectando falhas estruturais em um texto para indicação ao usuário de como melhorar sua qualidade (veja, por exemplo, Burstein et al., 2003; Feltrim et al., 2004);
- sistemas de diálogo, tratando-se o discurso sendo produzido, relacionando-se suas partes e interagindo com o usuário quando necessário (veja, por exemplo, Moore e Paris, 1993; Moore, 1995).

As análises produzidas pelo sistema também podem ser utilizadas para estudos de fenômenos lingüísticos e como estes se relacionam com a estrutura discursiva subjacente ao texto. Kehler (2002), por exemplo, explica vários fenômenos em função do discurso, como elipses, tempo verbal e orações coordenadas. As estruturas produzidas pelo DiZer poderiam ser a base de estudos dessa natureza.

6.1.4. Extensões

O DiZer é um primeiro passo para que outros aspectos do discurso sejam abordados. Como discutido no Capítulo 2 desta tese, o discurso pode ser representado em vários níveis, abordando princípios de estruturação e critérios distintos. Em geral, estes

níveis são representados por diferentes teorias discursivas. Entretanto, diante da possibilidade de mapeamento entre estas teorias, é possível que se gerem esses outros níveis.

Para a produção da estrutura semântica de um texto segundo a proposta de Kehler (2002), pode-se desenvolver um módulo de conversão das relações retóricas da RST para as 3 relações semânticas de Kehler. A partir da especificação de Kehler, uma possibilidade de mapeamento entre as relações é sugerida na Tabela 6.1.

Tabela 6.1 – Mapeamento das relações da RST nas relações semânticas de Kehler (2002)

Relações retóricas	**Relações semânticas**
ANTITHESIS	*Resemblance*
ATTRIBUTION	
BACKGROUND	
CIRCUMSTANCE	
COMPARISON	
CONCESSION	
CONCLUSION	
ELABORATION	
EVALUATION	
INTERPRETATION	
OTHERWISE	
PARENTHETICAL	
RESTATEMENT	
SUMMARY	
CONTRAST	
JOINT	
LIST	
SAME-UNIT	
CONDITION	*Cause-Effect*
ENABLEMENT	
EVIDENCE	
EXPLANATION	
JUSTIFY	
MEANS	
MOTIVATION	
NON-VOLITIONAL CAUSE	
NON-VOLITIONAL RESULT	
PURPOSE	
SOLUTIONHOOD	
VOLITIONAL CAUSE	
VOLITIONAL RESULT	
SEQUENCE	*Contiguity*

Para o mapeamento das relações retóricas para as relações semânticas de Jordan (1992), os algoritmos propostos por Korelsky e Kittredge (1993) podem ser

necessários, dado que o conjunto de relações de Jordan é mais sofisticado do que o de Kehler.

Para a produção das relações intencionais de Grosz e Sidner (1986) a partir da estrutura retórica de um texto, pode-se recorrer ao mapeamento sugerido por Moser e Moore (1996) e Marcu (2000a), segundo o qual as relações intencionais são determinadas a partir da nuclearidade das proposições relacionadas.

Os mapeamentos sugeridos entre os níveis do discurso podem ser desenvolvidos em módulos independentes que podem ser acoplados ao DiZer, enriquecendo sua saída e, desta forma, possibilitando seu uso em aplicações que exijam mais do que somente a estruturação retórica de um texto.

O DiZer também pode ser adaptado para a análise de textos de outros gêneros e domínios. Para isso, novos padrões de análise, representativos do corpus almejado, devem ser inseridos no repositório de padrões do DiZer. Em especial, devem ser desenvolvidos padrões de análise para as palavras e frases indicativas do novo tipo de texto sendo tratado, pois estes são, normalmente, dependentes de gênero e domínio textual. Com isso, as classes de informações necessárias para a aplicação dos novos padrões desenvolvidos devem ser inseridas no repositório de classes do DiZer. Como exemplo dessa necessidade, a frase indicativa "Os resultados deste trabalho são" não é usualmente encontrada em textos sobre esportes, supondo que estes sejam os novos textos para os quais se deseje automatizar a análise retórica. Esses textos contêm, por exemplo, frases como "O placar final foi". É interessante ressaltar que os marcadores discursivos, em oposição às palavras e frases indicativas, são, normalmente, utilizados uniforme e consistentemente em qualquer texto, não necessitando de adaptações.

6.1.5. Limitações

As limitações do DiZer são provenientes, em sua maioria, das restrições no processamento automático que são necessárias para a análise discursiva.

Durante o processo de segmentação do texto, para identificação das proposições simples que comporão as estruturas retóricas, delimitam-se somente os segmentos que contêm verbos. Essa estratégia garante uma segmentação mais consistente e com probabilidade menor de ocorrência de erros. Por outro lado, sabe-se que segmentos que não contêm verbos também podem expressar proposições simples.

Entretanto, a modelagem computacional de um processo que reconhecesse tais segmentos como válidos e descartasse os não válidos não é simples.

Em termos dos erros cometidos pelo DiZer durante a segmentação do texto, muitos se devem à não detecção das orações relativas, as quais deveriam ser relacionadas às proposições restantes do texto por relações encaixadas. No futuro, o uso de um parser deve resolver esse problema. Segundo Matthiessen e Thompson (1987), o uso de informação sintática pode, ainda, ajudar na determinação automática de quais proposições são núcleos e satélites das relações. Basicamente, os autores sugerem que orações subordinadas são satélites das relações retóricas.

No processo de detecção das relações retóricas entre proposições, somente relações entre proposições expressas por segmentos adjacentes são detectadas. Teoricamente, segundo a RST, as relações retóricas podem se estabelecer entre proposições expressas por segmentos distantes uns dos outros. A restrição da adjacência utilizada no DiZer garante um processamento mais eficiente. Entretanto, futuramente, este pode ser um possível tópico de pesquisa.

Alguns dos erros produzidos pelo DiZer durante a análise retórica de um texto são provenientes de erros cometidos pelo tagger utilizado, que tem precisão média de 89%. A utilização de um tagger com maior precisão ou de um módulo de pós-processamento para a correção da etiquetagem realizada podem melhorar os resultados do DiZer.

6.2. Sobre os Modelos Estatísticos

Os modelos estatísticos apresentados nesta tese de doutorado são o resultado da investigação de uma grande quantidade de modelos desenvolvidos, testados e aprimorados. Diferentemente da abordagem simbólica, na abordagem estatística, uma vez que a modelagem é definida, um modelo é rapidamente treinado e testado, com custo significativamente menor. Entretanto, o conhecimento produzido por estes modelos é muito dependente do método de aprendizado utilizado e do próprio formalismo dos modelos, sendo, portanto, de difícil interpretação por humanos. Além disso, por ser uma linha empírica de investigação, o desenvolvimento e aprimoramento dos modelos se dá, basicamente, por testes exaustivos, considerando-se diversos parâmetros e possibilidades.

Os modelos para análise discursiva, baseados em unidades de informação de crescente complexidade (palavras, conceitos e estruturas argumentais) apresentaram resultados promissores. São modelos inéditos aplicados a uma tarefa ainda não tratada desta forma anteriormente. Esses modelos foram treinados e testados para a língua inglesa e para uma única relação de discurso, a relação causa-efeito, similar à relação *cause-effect* de Kehler (2002).

O resultado dos testes dos modelos apontou a necessidade do aprendizado de estruturas argumentais, já que esta tarefa não foi satisfatoriamente cumprida pelo último dos modelos testados. Com isso, foi desenvolvido um modelo estatístico dedicado a este aprendizado, produzindo bons resultados. Esse modelo foi treinado com os 1.500 verbos mais freqüentes do inglês, resultando em um repositório chamado ArgBank.

6.2.1. Aplicações em PLN

Como Marcu e Popescu (2005) discutem, os modelos estatísticos têm sido a base de avanços significativos em PLN. Os autores mostram como esses modelos podem ser utilizados para a realização de inferências e de aprendizado de conhecimento de mundo. Acredita-se que os modelos apresentados nesta tese dão mais um passo nesta direção.

Em tradução automática estatística, a partir dos parâmetros probabilísticos aprendidos, pode-se (a) traduzir uma sentença em uma língua fonte para uma sentença em uma língua destino ou (b) reconhecer sentenças que são traduções umas das outras. O uso dos modelos estatísticos desenvolvidos neste trabalho refere-se ao item (b) acima, isto é, o uso dos parâmetros aprendidos para determinação das relações de discurso entre proposições. A aplicação do item (a), no âmbito deste trabalho, refere-se à possível tradução de uma causa em seus efeitos ou vice-versa, ou seja, a predição dos efeitos possíveis para uma dada causa ou vice-versa. Em um teste realizado, o modelo baseado em estruturas argumentais foi testado desta forma. Foi desenvolvido um decodificador (veja Seção 5.1 do capítulo anterior) que, a partir do conjunto de parâmetros de causa-efeito aprendido pelo método EM durante o treinamento do modelo baseado em estruturas argumentais (isto é, os parâmetros ce), procura pelos efeitos mais prováveis dada uma causa ou vice-versa. Por exemplo, dada a estrutura

argumental de causa *was*(*he*,*insane*), o decodificador sugere como possível efeito *argue*(*lawyers*,*innocence*).

Em uma avaliação subjetiva, por um único lingüista computacional, obtiveram-se os seguintes resultados para a predição de efeitos: em 40% dos casos, as predições faziam sentido, consideradas passíveis de ocorrência no mundo real; em 25% dos casos, não foi possível determinar com certeza se as predições eram plausíveis; em 35% dos casos, os efeitos previstos não faziam sentido. Nos casos em que não foi possível determinar se os efeitos eram plausíveis ou não, dois problemas foram identificados nas estruturas argumentais sugeridas: presença de nomes próprios e presença de palavras estranhas ou desconhecidas, como ocorrem nas estruturas *killed*(*capano*,*fahey*) e *shut*(*camp*,*unher*). Acredita-se que, melhorando-se a qualidade dos pares de causa-efeito aprendidos pelo modelo baseado em estruturas argumentais, os resultados da predição devem melhorar também. Como pesquisa futura, isso pode ocorrer com o uso do modelo de aprendizado de estruturas argumentais proposto.

O repositório de estruturas argumentais produzido pode ser utilizado para diversos fins em PLN. Uma utilização natural desse repositório é no auxílio à interpretação semântica de textos, pela identificação da estrutura argumental mais provável em uma sentença, por exemplo. Em aplicações de geração automática de textos, o repositório pode servir como indicador da qualidade do texto gerado: caso alguma estrutura do repositório seja observada no texto, pode haver grandes chances do texto ser semanticamente bem formado; caso contrário, pode-se refazer o texto.

6.2.2. Extensões

Os modelos apresentados podem ser treinados para todas as relações do discurso que se deseje e para diferentes línguas. O treinamento para outras relações é possível devido à generalidade das modelagens propostas. O uso dos modelos para outras línguas depende, entretanto, da disponibilidade dos recursos e ferramentas de PLN necessários.

Repositórios de conhecimento podem ser acoplados aos modelos, tornando-os mais informados para a tarefa que realizam. Por exemplo, os modelos de análise discursiva, quando aplicados às relações causa-efeito, podem assumir como conhecimento pré-definido ou complementar as relações de causa entre os verbos

previstas na WordNet. Por exemplo, para o verbo *kill* (“matar”, em português), especifica-se, na WordNet, os efeitos *die* e *pass away* (“morrer”, em português), entre outros.

O modelo desenvolvido para o aprendizado das estruturas argumentais dos verbos deverá, no futuro, ser aplicado ao português do Brasil, produzindo um repositório similar ao ArgBank. Diferentemente da língua inglesa, para a qual há repositórios como a FrameNet, a VerbNet e o PropBank, para o português não há nenhum.

6.2.3. Limitações

Os modelos estatísticos propostos podem ser aprimorados para serem capazes de lidar com sentenças longas e que contenham construções lingüísticas complexas.

Em seu estágio atual, o modelo para o aprendizado das estruturas argumentais dos verbos é treinado com sentenças de, no máximo, 10 palavras, e restringe a 3 o número de argumentos possíveis para um verbo. Além disso, o modelo não é capaz de detectar *phrasal verbs* e tratá-los adequadamente.

Essas limitações constituem trabalhos futuros para o aprimoramento dos modelos.

6.3. Considerações Finais

O uso das estruturas do ArgBank para o aprendizado de pares de estruturas de causa-efeito, aprimorando o modelo para análise discursiva baseado em estruturas argumentais, constitui o próximo passo da pesquisa relatada nesta tese. Os resultados deste aprendizado deverão servir de base para outras aplicações, por exemplo, a predição de efeitos para causas.

Neste trabalho de doutorado, pela investigação das abordagens simbólica e estatística para a realização da análise discursiva automática, mostra-se como conhecimentos de naturezas diversas podem contribuir para esta tarefa. Foram abordados os marcadores discursivos, que constituem o principal mecanismo lingüístico para a detecção de relações retóricas; palavras e frases indicativas, usualmente dependentes de gênero e domínio textual, foram investigadas; estatísticas

sobre a organização discursiva foram utilizadas para o ranqueamento das estruturas possíveis para um mesmo texto; o relacionamento entre palavras e conceitos para a análise discursiva foi investigado; verificou-se como o relacionamento entre as estruturas argumentais subjacentes aos segmentos textuais pode ser utilizado para a automação da análise proposta. Nestes dois últimos casos, em particular, tem-se o aprendizado de parâmetros probabilísticos que codificam regras semânticas para a análise discursiva, por exemplo, o conjunto de parâmetros ce dos modelos estatísticos.

Acredita-se que a modelagem estatística constitui a principal linha de investigação para a automação da análise discursiva no futuro próximo, por ser generalizável, adaptável e treinável. Entretanto, diante do relato de Marcu e Popescu (2005) e da percepção obtida com o desenvolvimento deste trabalho de doutorado, vê-se que muito trabalho ainda é necessário para o completo entendimento de como tratar o problema em questão para que se produzam analisadores discursivos automáticos com desempenho satisfatório.

As ferramentas e recursos produzidos por este trabalho de doutorado encontram-se disponíveis para uso pela comunidade de pesquisa na *webpage* do NILC, grupo no qual este trabalho foi desenvolvido. Espera-se que a pesquisa relatada nesta tese possa impulsionar as pesquisas em análise discursiva automática, principalmente para o português do Brasil, avançando o estado da arte em PLN.

Referências

Aires, R.V.X.; Aluísio, S.M.; Kuhn, D.C.S.; Andreeta, M.L.B.; Oliveira Jr., O.N. (2000). Combining Multiple Classifiers to Improve Part of Speech Tagging: A Case Study for Brazilian Portuguese. In the *Proceedings of the Brazilian AI Symposium* (SBIA'2000), pp. 20-22.

Baker, C.F.; Fillmore, C.J.; Lowe, J.B. (1998). The Berkeley FrameNet project. In the *Proceedings of COLING/ACL*, pp. 86-90, Montreal.

Bikel, D.M.; Schwartz, R.; Weischedel, R.M. (1999). An Algorithm that Learns What's in a Name. *Machine Learning* (Special Issue on NLP).

Brent, M.R. (1991). Automatic acquisition of subcategorization frames from untagged text. In the *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 209-214, Berkeley, CA.

Briscoe, T. and Carroll, J. (1997). Automatic extraction of subcategorization from corpora. In the *Proceedings of the 5th ANLP Conference*, pp. 356-363, Washington, D.C.

Brown, P.; Cocke, J.; Della Pietra, S.; Della Pietra, V.; Jelinek, F.; Lafferty, J.; Mercer, R.; Roossin, P. (1990). A statistical approach to machine translation. *Computational Linguistics*, Vol. 16, N. 2, pp. 79-85.

Brown, P.; Della Pietra, S.; Della Pietra, V.; Mercer, R. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, Vol. 19, N. 2, pp. 263-311.

Burstein, J.; Marcu, D.; Knight, K. (2003). Finding the WRITE Stuff: Automatic Identification of Discourse Structure in Student Essays. *IEEE Intelligent Systems*, pp. 32-39.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *Computational Linguistics*, Vol. 22, N. 2, pp. 249–254.

Carlson, L. and Marcu, D. (2001). *Discourse Tagging Reference Manual*. ISI Technical Report ISI-TR-545.

Carlson, L.; Marcu, D.; Okurowski, M.E. (2002). Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In J. van Kuppevelt and R. Smith (eds.), *Current Directions in Discourse and Dialogue*, Kluwer Academic Publishers.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. PhD Thesis, University of Pennsylvania.

Corston-Oliver, S. (1998). *Computing Representations of the Structure of Written Discourse*. PhD Thesis, University of California, Santa Barbara, CA, USA.

Cristea, D.; Ide, N.; Romary, L. (1998): Veins Theory. An Approach to Global Cohesion and Coherence. In the *Proceedings of Coling/ACL*.

Dale, R. (1993). Rhetoric and Intentions in Discourse. In the *Proceedings of the Intentionality and Structure in Discourse Relations Workshop*, pp. 5-6. Ohio, USA.

Daumé, H. and Marcu, D. (2004). A Phrase-Based HMM Approach to Document/Abstract Alignment. In the *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Dempster, A.P.; Laird, N.M.; Rugin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, Series B, Vol. 39, pp. 1-38.

Di Eugenio, B. (1992). Understanding natural language instructions: the case of purpose clauses. In the *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL'92)*, pp. 120-127. Newmark, DE.

Di Eugenio, B. (1993). *Understanding natural language instructions: a computational approach to purpose clauses*. Ph.D. thesis, University of Pennsylvania.

Dias da Silva, B.C. (1996). *A face tecnológica dos estudos da linguagem: o processamento automático das línguas naturais*. Tese de Doutorado. Faculdade de Ciências e Letras, Universidade Estadual Paulista – UNESP, Araraquara.

Dias da Silva, B.C. e Oliveira, M.F. (2002). Inclusão de informação pragmático-discursiva na base lexical de um thesaurus eletrônico. *Estudos Lingüísticos*, Vol. 31.

Elhadad, M, and McKewon, K. R. (1990). Generating connectives. In the *Proceedings of the International Conference on Computational Linguistics (COLING'90)*, Vol. 3, pp. 97-102. Helsinki, Finland.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. The MIT Press.

Feltrim, V.D.; Nunes, M.G.V.; Aluísio, S.M. (2001). *Um corpus de textos científicos em Português para a análise da Estrutura Esquemática*. Série de Relatórios do NILC. NILC-TR-01-4.

Feltrim, V.D.; Aluísio, S.M.; Nunes, M.G.V. (2003). Analysis of the rhetorical structure of computer science abstracts in Portuguese. In D. Archer, P. Rayson, A. Wilson and T. McEnery (eds.), *Proceedings of the Corpus Linguistics*, UCREL Technical Papers, Vol. 16, Part 1, pp. 212-218.

Feltrim, V.D.; Pelizzoni, J.M.; Teufel, S.; Nunes, M.G.V.; Aluísio, S.M. (2004). Applying Argumentative Zoning in an Automatic Critiquer of Academic Writing. In the *Proceedings of the XVII Brazilian Symposium on Artificial Intelligence*, pp. 214-223.

Framis, F.R. (1994). An experiment on learning appropriate selection restrictions from a parsed corpus. In the *Proceedings of the International Conference on Computational Linguistics*, Kyoto, Japan.

Fraser, B. (1999). What are discourse markers? *Journal of Pragmatics*, Vol. 32, pp. 913-952.

Gildea, D. (2002). Probabilistic Models of Verb-Argument Structure. In the *Proceedings of the 17th International Conference on Computational Linguistics*.

Gomez, F. (2004). Building Verb Predicates: A Computational View. In the *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pp. 359-366, Barcelona, Spain.

Green, R.; Dorr, B.J.; Resnik, P. (2004). Inducing Frame Semantic Verb Classes from WordNet and LDOCE. In the *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pp. 375-382, Barcelona, Spain.

Grishman, R. and Sterling, J. (1992). Acquisition of selectional patterns. In the *Proceedings of the International Conference on Computational Linguistics*, pp. 658-664, Nantes, France.

Grishman, R. and Sterling, J. (1994). Generalizing Automatically Generated Selectional Patterns. In the *Proceedings of the 15th International Conference on Computational Linguistics*, Kyoto, Japan.

Grosz, B.; Joshi, A.; Weisten, S. (1995). Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, V. 21, N. 2, pp. 203-225.

Grosz, B. and Sidner, C. (1986). Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, Vol. 12, N. 3.

Grote, B.; Lenke, N.; Stede, M. (1997). Ma(r)king concessions in English and German. *Discourse Processes*, pp. 87-117.

Hanneforth, T.; Heintze, S.; Stede, M. (2003). Rhetorical Parsing with Underspecification and Forests. In the *Proceedings of HLT-NAACL*.

Hearst, M. (1997). TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *Computational Linguistics*, Vol. 23, N. 1, pp. 33-64.

Hirschberg, J. and Litman, D. J. (1987). Now lets's talk about *now*: identifying cue phrases intonationally. In the *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL-87)*, pp. 163-171. Stanford, CA.

Hirschberg, J. and Litman, D. J. (1993). Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, Vol. 19, N. 3, pp. 501-513.

Hoey, M.P. (1979). *Signalling in Discourse*. University of Birmingham, England.

Hoey, M.P. (1983a). *On the Surface of Discourse*. London: George Allen and Unwin.

Hoey, M.P. (1983b). The place of clause relational analysis in linguistic description. *English Language Research Journal*, N. 4.

Hoey, M.P. and Winter, E.O. (1986). Clause relations and the writer's communicative task. In B. Couture (ed.), *Functional Approaches to Writing*. London: Frances Pinter.

Hovy, E. (1988). *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum Associates Publishers, Hillsdale, New Jersey.

Hovy, E. (1991). Approaches to the planning of coherent text. In C. Paris, W. Swartout and W. Mann (eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 83-102. Kluwer Academic Publishers, Boston.

Hovy, E. (1993). In Defense of Syntax: Informational, Intentional, and Rhetorical Structures in Discourse. In the *Proceedings of the Intentionality and Structure in Discourse Relations Workshop*, pp. 35-39. Ohio, USA.

Jordan, M.P. (1978). *The principal semantics of the nominals 'this' and 'that' in contemporary English writing*. PhD Thesis, The Hatfield Polytechnic and Birmingham University, England.

Jordan, M.P. (1980). Short Texts to Explain Problem-Solution Structures – and Vice Versa. *Instructional Science*, Vol. 9, pp. 221-252

Jordan, M.P. (1984). Structure, style and word choice in everyday English texts. *TESL* 15, Vols. 1 & 2.

Jordan, M.P. (1985a). Some clause relational associated nominals in technical English. *Technostyle*, Vol. 4, N. 1.

Jordan, M.P. (1985b). Some relations of surprise and expectation in English. In J. Hall (ed.), *The 11th LACUS Forum*. Columbia SC: Hornbeam Press.

Jordan, M.P. (1988). Some advances in clause relational theory. In J.D. Benson and W.S. Greaves (eds.), *Systemic Functional Approaches to Discourse*. Norwood NJ: Ablex.

Jordan, M.P. (1989). Relational propositions within the clause. In S. Embleton (ed.), *The 15th LACUS Forum*. Lake Bluffs IL: LACUS.

Jordan, M.P. (1992). An Integrated Three-Pronged Analysis of a Fund-Raising Letter. In W.C. Mann and S.A. Thompson (eds.), *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pp. 171-226.

Jurafsky, D. and Martin, J.H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Kehler, A. (2002). *Coherence, Reference and the Theory of Grammar*. CSLI Publications.

Kingsbury, P. and Palmer, M. (2002). From Treebank to PropBank. In the *Proceedings of the 3rd International Conference on Language Resources and Evaluation*, Las Palmas.

Kipper, K.; Dang, H.T.; Palmer, M. (2000). Class-based Construction of a Verb Lexicon. In the *Proceedings of AAAI 17th National Conference on Artificial Intelligence*. Austin, Texas.

Knott, A. (1995). *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. Ph.D. thesis, University of Edinburgh, Scotland.

Knott, A. and Dale, R. (1996). Choosing a set of coherence relations for text generation: a data-driven approach. In M. Zock (ed), *Trends in Natural Language Generation: an Artificial Intelligence Perspective*, pp. 47-67. Heidelberg, Germany.

Knott, A. and Mellish, C. (1996). A feature-based account of the relations signaled by sentence and clause connectives. *Journal of Language and Speech*, Vol. 39, Ns. 2 and 3, pp. 143-183.

Koch, I.V. (1998). *A Coesão Textual*. Editora Contexto.

Koch, I.V. e Travaglia, L.C. (2002). *A Coerência Textual*. Editora Contexto.

Koehn, P.; Och, F.J.; Marcu, D. (2003). Statistical Phrase-Based Translation. In the *Proceedings of the Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics*.

Korelsky, T. and Kittredge, R. (1993). Towards stratification of RST. In the *Proceedings of the Intentionality and Structure in Discourse Relations Workshop*, pp. 52-55. Ohio, USA.

Korhonen, A. (2002). Semantically Motivated Subcate-gorization Acquisition. In the *Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon*, pp. 51-58.

Lapata, M. (1999). Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In the *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 394-404.

Maier, E. (1993). The Representation of Interdependencies between Communicative Goals and Rhetorical Relations in the Framework of Multimedia Document Generation. In the *Proceedings of the Intentionality and Structure in Discourse Relations Workshop*, pp. 70-73. Ohio, USA.

Maier, E. and Hovy, E. (1991). A Metafunctionally Motivated Taxonomy for Discourse Structure Relations. In the *Proceedings of the 3rd European Workshop on Language Generation*. Innsbruck, Austria.

Mahmud, R. and Ramsay, A. (2005). Finding Discourse Relations in Student Essays. In the *Proceedings of the 6th Computational Linguistics and Intelligent Text Processing International Conference*.

Manning, C.D. (1993). Automatic acquisition of a large subcategorization dictionary from corpora. In the *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 235-242, Columbus, Ohio.

Manning, C.D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

Mann, W.C. and Thompson, S.A. (1987). *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report ISI/RS-87-190.

Marcu, D. (1997). *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*. PhD Thesis, Department of Computer Science, University of Toronto.

Marcu, D. (1999). A formal and computational synthesis of Grosz and Sidner's and Mann and Thompson's theories. In the *Proceedings of the Workshop on Levels of Representation in Discourse*, pp. 101-108. Edinburgh, Scotland.

Marcu, D. (2000a). Extending a Formal and Computational Model of Rhetorical Structure Theory with Intentional Structures à la Grosz and Sidner. In the *Proceedings of the 18th International Conference on Computational Linguistics (COLING'2000),* Saarbrueken.

Marcu, D. (2000b). *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press. Cambridge, Massachusetts.

Marcu, D.; Carlson, L.; Watanabe, M. (2000). The Automatic Translation of Discourse Structures. In the *Proceedings of the 1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics*. Seattle, Washington.

Marcu, D. and Echihabi, A. (2002). An Unsupervised Approach to Recognizing Discourse Relations. In the *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL'02), Philadelphia, PA.

Marcu, D. and Popescu, A.M. (2005). Towards Developing Probabilistic Generative Models for Reasoning with Natural Language Representations. In the *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 88-99.

Martins, R.T.; Hasegawa, R.; Nunes, M.G.V.; Montilha, G.; Oliveira Jr., O.N. (1998). Linguistic issues in the development of ReGra: a Grammar Checker for Brazilian Portuguese. *Natural Language Engineering*, Vol. 4, pp. 287-307. Cambridge University Press.

Matthiessen, C.M.I.M. and Thompson, S.A. (1987). The Structure of Discourse and "Subordination". In J. Haiman and S.A. Thompson (eds.), *Clause Combining in Discourse and Grammar*. Amsterdam, John Benjamins.

Maybury, M.T. (1992). Communicative Acts for Explanation Generation. *Int. Journal of Man-Machine Studies 37*, pp. 135-172.

McCarthy, D. (2000). Using semantic preferences to identify verbal participation in role switching alternations. In the *Proceedings of the 1st NAACL*, pp. 256-263, Seattle, Washington.

Merlo, P. and Stevenson, S. (2001). Automatic Verb Classification Based on Statistical Distributions of Argument Structure. *Computational Linguistics*, Vol. 27, N. 3.

Mitchell, T.M. (1997). *Machine Learning*. McGraw Hill, New York.

Moore, J.D. (1995). *Participating in Explanatory Dialogs: Interpreting and Responding to Questions in Context*. The MIT Press. Cambridge, Massachusetts.

Moore, J.D. and Paris, C. (1993). Plannig Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. *Computational Linguistics*, Vol. 19, N. 4, pp. 651-694.

Moore, J. D. and Pollack, M. E. (1992). A problem for RST: the need for multi-level discourse analysis. *Computational Linguistics*, Vol. 18, N. 4, pp. 537-544.

Moser, M. and Moore, J. D. (1996). Toward a synthesis of two accounts of discourse structure. *Computational Linguistics*, Vol. 22, N. 3, pp. 409-419.

O'Donnell, M. (1997). Variable-Length On-Line Document Generation. In the *Proceedings of the 6th European Workshop on Natural Language Generation*, Gerhard-Mercator University, Duisburg, Germany.

Oates, S.L. (1999). *State of the Art Reporto n Discourse Markers and Relations*. Technical Report ITRI-99-08. Information Technology Research Institute. University of Brighton.

Paice, C.D. (1981). The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. *Information Retrieval Research*. Butterworth & Co. (Publishers).

Paizan, D.C. (2001). *O uso da linguagem da Internet na produção de um módulo de ensino de leitura de inglês instrumental*. Dissertação de Mestrado. Faculdade de Ciências e Letras de Araraquara.

Pardo, T.A.S. (2002). *DMSumm: Um Gerador Automático de Sumários*. Dissertação de Mestrado. Departamento de Computação. Universidade Federal de São Carlos. São Carlos – SP.

Pardo, T.A.S. e Nunes, M.G.V. (2003a). *Análise de Discurso: Teorias Discursivas e Aplicações em Processamento de Línguas Naturais*. Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação - ICMC, Universidade de São Paulo, no. 196.

Pardo, T.A.S. e Nunes, M.G.V. (2003b). *Segmentação Textual Automática: Uma Revisão Bibliográfica*. Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação - ICMC, Universidade de São Paulo, no. 185.

Pardo, T.A.S. e Nunes, M.G.V. (2004). *Relações Retóricas e seus Marcadores Superficiais: Análise de um Corpus de Textos Científicos em Português do Brasil*. Série de Relatórios Técnicos do Instituto de Ciências Matemáticas e de Computação - ICMC, Universidade de São Paulo, no. 231.

Pardo, T.A.S. and Rino, L.H.M. (2002). DMSumm: Review and Assessment. In E. Ranchhod and N. J. Mamede (eds.), *Advances in Natural Language Processing*, pp. 263-273 (Lecture Notes in Artificial Intelligence 2389). Springer-Verlag, Germany.

Pinheiro, G.M. e Aluísio, S.M. (2003). *Corpus NILC: Descrição e Análise Crítica com Vistas ao Projeto Lacio-Web*. Série de Relatórios Técnicos do Instituto de Ciencias Matemáticas e de Computação – ICMC, Universidade de São Paulo, N. 190.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Quirk, R.; Greenbaum, S.; Leech, G.; Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman, Harcourt.

Ratnaparkhi, A. (1996). A Maximum Entropy Part-of-Speech Tagger. In the *Proceedings of the 1st Empirical Methods in Natural Language Processing Conference*. Philadelphia.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge, University Press.

Reitter, D. (2003). Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. *GLDV-Journal for Computational Linguistics and Language Technology*, Vol. 18, pp. 38-52.

Reitter, D. and Stede, M. (2003). Step by step: underspecified markup in incremental rhetorical analysis. In the *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora*.

Resnik, P. (1992). Wordnet and distributional analysis: a class-based approach to lexical discovery. In the *Proceedings of AAAI Workshop on Statistical Methods in NLP*.

Rino, L.H.M. (1996). *Modelagem de Discurso para o Tratamento da Concisão e Preservação da Idéia Central na Geração de Textos*. Tese de Doutorado. IFSC-Usp. São Carlos - SP.

Roman, N.T. and Carvalho, A.M.B.R. (2002). Task Oriented Dialog Processing Using Multiagents Theory. In F.J. Garijo, J.C. Riquelme and M. Toro (eds.), *Proceedings of the 8th Ibero-American Conference on Artificial Intelligence*, pp. 704-713. Seville, Spain.

Rooth, M.; Stefan, R.; Prescher, D.; Carroll, G.; Beil, F. (1999). Inducing a semantically annotated lexicon via EM-based clustering. In the *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 104-111, College Park, Maryland.

Russel, S.J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

Salton, G. (1971). *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall, New York.

Sarkar, A. and Zeman, D. (2000). Automatic extraction of subcategorization frames for Czech. In the *Proceedings of the 18th International Conference on Computational Linguistics*.

Sarkar, A. and Tripasai, W. (2002). Learning Verb Argument Structures from Minimally Annotated Corpora. In the *Proceedings of the 19th International Conference on Computational Linguistics*.

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal*, Vol. 27, N. 3, pp. 379-423.

Schilder, F. (2002). Robust discourse parsing via discourse markers, topicality and position. In J. Tait, B.K. Boguraev and C. Jacquemin (eds.), *Natural Language Engineering*, Vol. 8. Cambridge University Press.

Seno, E.R.M. e Rino, L.H.M. (2005). *Heurísticas de Sumarização de Estruturas RST*. Série de Relatórios do NILC. NILC-TR-05-04.

Soricut, R. and Brill, E. (2004). Automatic Question Answering: Beyond the Factoid. In the *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*.

Soricut, R. and Marcu, D. (2003). Sentence Level Discourse Parsing using Syntactic and Lexical Information. In the *Proceedings of HLT/NAACL*.

Vapnik, V.N. (1995). *The nature of statistical learning theory*. New York: Springer.

Voorhees, E.M. and Buckland, L.P. (eds.) (2002). *NIST Special Publication 500-251: The Eleventh Text REtrieval Conference* (TREC 2002). Department of Commerce, National Institute of Standards and Technology.

Williams, S. and Reiter, E. (2003). A corpus analysis of discourse relations for Natural Language Generation. In the *Proceedings of Corpus Linguistics*, pp. 899-908. Lancaster University.

Winter, E.O. (1968). Some aspects of cohesion. In R.D. Huddleston, R.A. Hudson., E.O. Winter and A. Henrici (eds.), *Sentence and Clause in Scientific English*. University of London.

Winter, E.O. (1971). *Connection in science material: A proposition about the semantics of clause relations*. Centre for Information on English Language Teaching and Research.

Winter, E.O. (1974). *Replacement as a function of repetition*. PhD Thesis, University of London.

Winter, E.O. (1976). *Fundamentals of Information Structure*. Hatfield Polytechnic, Hertfodshire, England.

Winter, E.O. (1977). A Clause-Relational Approach to English Texts. A Study of Some Predictive Lexical Items in Written Discourse. *Structional Science*, Vol. 6, N. 1, pp. 1-92.

Winter, E.O. (1979). Replacement as a Fundamental Function of the Sentence in Context. In *Forum Linguitics*, Vol. 4, N. 2, pp. 95-133.

Winter, E.O. (1982). *Towards a Contextual Grammar of English*. London: George Allen and Unwin.

Witten, I.H. and Frank, E. (2000). *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementation*. Morgan Kaufmann.

Apêndice A – Definição das Relações Retóricas

Neste apêndice, são apresentadas as definições das relações retóricas utilizadas neste trabalho de doutorado. Na Tabela A.1, mostram-se a lista completa das relações, identificando-se as relações multinucleares, e a natureza das relações. A seguir, nas Figuras A.1-A.32, mostram-se as definições das relações.

Tabela A.1 – Elenco de relações retóricas

Relações	Multinuclear	Natureza das relações
ANTITHESIS	Não	Intencional
ATTRIBUTION	Não	Estrutural
BACKGROUND	Não	Intencional
CIRCUMSTANCE	Não	Semântica
COMPARISON	Não	Semântica
CONCESSION	Não	Intencional
CONCLUSION	Não	Semântica
CONDITION	Não	Semântica
ELABORATION	Não	Semântica
ENABLEMENT	Não	Intencional
EVALUATION	Não	Semântica
EVIDENCE	Não	Intencional
EXPLANATION	Não	Semântica
INTERPRETATION	Não	Semântica
JUSTIFY	Não	Intencional
MEANS	Não	Semântica
MOTIVATION	Não	Intencional
NON-VOLITIONAL CAUSE	Não	Semântica
NON-VOLITIONAL RESULT	Não	Semântica
OTHERWISE	Não	Semântica
PARENTHETICAL	Não	Estrutural
PURPOSE	Não	Semântica
RESTATEMENT	Não	Semântica
SOLUTIONHOOD	Não	Semântica
SUMMARY	Não	Semântica
VOLITIONAL CAUSE	Não	Semântica
VOLITIONAL RESULT	Não	Semântica
CONTRAST	Sim	Semântica
JOINT	Sim	Semântica
LIST	Sim	Semântica
SAME-UNIT	Sim	Estrutural
SEQUENCE	Sim	Semântica

Nome da relação: ANTITHESIS
Restrições sobre N: o escritor julga N válido **Restrições sobre S**: não há **Restrições sobre N+S**: N e S estão em contraste; por causa da aparente incompatibilidade, não se pode julgar N e S válidos ao mesmo tempo; a compreensão de S e da incompatibilidade entre N e S faz o leitor aceitar melhor N **Efeito**: o leitor aceita melhor N

Figura A.1 – Definição da relação ANTITHESIS

Nome da relação: ATTRIBUTION
Restrições sobre N: N apresenta uma expressão, fala ou pensamento de alguém ou algo **Restrições sobre S**: S apresenta alguém ou algo que produz N **Restrições sobre N+S**: S e N indicam, respectivamente, a fonte de uma mensagem e a mensagem **Efeito**: o leitor é informado sobre a mensagem e sobre quem ou o que a produziu

Figura A.2 – Definição da relação ATTRIBUTION

Nome da relação: BACKGROUND
Restrições sobre N: o leitor não compreenderá suficientemente N antes de ler S **Restrições sobre S**: não há **Restrições sobre N+S**: S aumenta a habilidade do leitor em compreender algum elemento em N **Efeito**: a habilidade do leitor para compreender N aumenta

Figura A.3 – Definição da relação BACKGROUND

Nome da relação: CIRCUMSTANCE
Restrições sobre N: não há **Restrições sobre S**: apresenta uma situação (realizável) **Restrições sobre N+S**: S provê uma situação na qual o leitor pode interpretar N **Efeito**: o leitor reconhece que S provê uma situação na qual N deve ser interpretado

Figura A.4 – Definição da relação CIRCUMSTANCE

Nome da relação: COMPARISON
Restrições sobre N: apresenta uma característica de algo ou alguém **Restrições sobre S**: apresenta uma característica de algo ou alguém comparável com o que é apresentado em N **Restrições sobre N+S**: as características de S e N estão em comparação **Efeito**: o leitor reconhece que S é comparado a N em relação a certas características

Figura A.5 – Definição da relação COMPARISON

Nome da relação: CONCESSION
Restrições sobre N: o escritor julga N válido **Restrições sobre S**: o escritor não afirma que S pode não ser válido **Restrições sobre N+S**: o escritor mostra uma incompatibilidade aparente ou em potencial entre N e S; o reconhecimento da compatibilidade entre N e S melhora a aceitação de N pelo leitor **Efeito**: o leitor aceita melhor N

Figura A.6 – Definição da relação CONCESSION

Nome da relação: CONCLUSION
Restrições sobre N: não há **Restrições sobre S**: S baseia-se no que é apresentado em N **Restrições sobre N+S**: S apresenta um fato concluído a partir da interpretação de N **Efeito**: o leitor reconhece que S é uma conclusão produzida devido à interpretação de N

Figura A.7 – Definição da relação CONCLUSION

Nome da relação: CONDITION
Restrições sobre N: não há **Restrições sobre S**: S apresenta uma situação hipotética, futura ou não realizada **Restrições sobre N+S**: a realização de N depende da realização de S **Efeito**: o leitor reconhece como a realização de N depende da realização de S

Figura A.8 – Definição da relação CONDITION

Nome da relação: ELABORATION
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta detalhes adicionais sobre a situação ou algum elemento de N **Efeito**: o leitor reconhece S como apresentando detalhes adicionais sobre N

Figura A.9 – Definição da relação ELABORATION

Nome da relação: ENABLEMENT
Restrições sobre N: apresenta uma ação do leitor não realizada **Restrições sobre S**: não há **Restrições sobre N+S**: a compreensão de S pelo leitor aumenta sua habilidade para realizar a ação em N **Efeito**: a habilidade do leitor para realizar a ação em N aumenta

Figura A.10 – Definição da relação ENABLEMENT

Nome da relação: EVALUATION
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: S se relaciona a N pelo grau de avaliação positiva do escritor por N **Efeito**: o leitor reconhece que S avalia N e reconhece o valor que ele atribui

Figura A.11 – Definição da relação EVALUATION

Nome da relação: EVIDENCE
Restrições sobre N: o leitor poderia não acreditar em N de forma satisfatória para o escritor **Restrições sobre S**: o leitor acredita em S ou o achará válido **Restrições sobre N+S**: a compreensão de S pelo leitor aumenta sua convicção em N **Efeito**: a convicção do leitor em N aumenta

Figura A.12 – Definição da relação EVIDENCE

Nome da relação: EXPLANATION
Restrições sobre N: apresenta um evento ou situação **Restrições sobre S**: não há **Restrições sobre N+S**: S explica como e/ou porque o evento ou situação apresentado em N ocorre ou veio a ocorrer **Efeito**: o leitor reconhece que S é a razão para N ou que S explica como N ocorre

Figura A.13 – Definição da relação EXPLANATION

Nome da relação: INTERPRETATION
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta um conjunto de idéias que não é expresso em N propriamente, mas derivado deste **Efeito**: o leitor reconhece que S apresenta um conjunto de idéias que não é propriamente expresso no conhecimento fornecido por N

Figura A.14 – Definição da relação INTERPRETATION

Nome da relação: JUSTIFY
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: a compreensão de S pelo leitor aumenta sua prontidão para aceitar o direito do escritor de apresentar N **Efeito**: a prontidão do leitor para aceitar o direito do escritor de apresentar N aumenta

Figura A.15 – Definição da relação JUSTIFY

Nome da relação: MEANS
Restrições sobre N: uma atividade **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta um método ou instrumento que faz com que a realização de N seja mais provável **Efeito**: o leitor reconhece que o método ou instrumento em S faz com que a realização de N seja mais provável

Figura A.16 – Definição da relação MEANS

Nome da relação: MOTIVATION
Restrições sobre N: uma ação volitiva não realizada **Restrições sobre S**: não há **Restrições sobre N+S**: a compreensão de S motiva a realização de N **Efeito**: o leitor reconhece que S motiva a realização de N

Figura A.17 – Definição da relação MOTIVATION

Nome da relação: NON-VOLITIONAL CAUSE
Restrições sobre N: apresenta uma ação não volitiva **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta uma situação que pode ter causado N; sem S, o leitor poderia não reconhecer o que causou a ação em N; N é mais central para a satisfação do objetivo do escritor do que S **Efeito**: o leitor reconhece a situação apresentada em S como a causa da ação apresentada em N

Figura A.18 – Definição da relação NON-VOLITIONAL CAUSE

Nome da relação: NON-VOLITIONAL RESULT
Restrições sobre N: não há **Restrições sobre S**: apresenta uma ação não volitiva **Restrições sobre N+S**: N apresenta uma situação que pode ter causado S; sem N, o leitor poderia não reconhecer o que causou a ação em S; N é mais central para a satisfação do objetivo do escritor do que S **Efeito**: o leitor reconhece a situação apresentada em N como a causa da ação apresentada em S

Figura A.19 – Definição da relação NON-VOLITIONAL RESULT

Nome da relação: OTHERWISE
Restrições sobre N: apresenta uma situação não realizada **Restrições sobre S**: apresenta uma situação não realizada **Restrições sobre N+S**: a realização de N impede a realização de S **Efeito**: o leitor reconhece que a realização de N impede a realização de S

Figura A.20 – Definição da relação OTHERWISE

Nome da relação: PARENTHETICAL
Restrições sobre N: não há **Restrições sobre S**: apresenta informação extra relacionada a N que não está expressa no fluxo principal do texto **Restrições sobre N+S**: S apresenta informação extra relacionada a N, complementado N; S não pertence ao fluxo principal do texto **Efeito**: o leitor reconhece que S apresenta informação extra relacionada a N, complementando N

Figura A.21 – Definição da relação PARENTHETICAL

Nome da relação: PURPOSE
Restrições sobre N: apresenta uma ação **Restrições sobre S**: apresenta uma situação não realizada **Restrições sobre N+S**: S apresenta uma situação que pode realizar N **Efeito**: o leitor reconhece que a atividade em N pode ser iniciada por meio de S

Figura A.22 – Definição da relação PURPOSE

Nome da relação: RESTATEMENT
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: S se relaciona a N; ambos apresentam conteúdo comparável; N é mais importante para a satisfação do objetivo do escritor **Efeito**: o leitor reconhece que S expressa o mesmo conteúdo de N, mas de forma diferente

Figura A.23 – Definição da relação RESTATEMENT

Nome da relação: SOLUTIONHOOD
Restrições sobre N: não há **Restrições sobre S**: apresenta um problema **Restrições sobre N+S**: N é uma solução para o problema em S **Efeito**: o leitor reconhece N como uma solução para o problema em S

Figura A.24 – Definição da relação SOLUTIONHOOD

Nome da relação: SUMMARY
Restrições sobre N: não há **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta o conteúdo de N resumido **Efeito**: o leitor reconhece S como um resumo do conteúdo de N

Figura A.25 – Definição da relação SUMMARY

Nome da relação: VOLITIONAL CAUSE
Restrições sobre N: apresenta uma ação volitiva ou uma situação que poderia surgir de uma ação volitiva **Restrições sobre S**: não há **Restrições sobre N+S**: S apresenta uma situação que pode ter acarretado o fato do agente da ação volitiva em N ter realizado a ação; sem S, o leitor poderia não reconhecer a motivação da ação; N é mais central para a satisfação do objetivo do escritor do que S **Efeito**: o leitor reconhece a situação apresentada em S como a causa da ação apresentada em N

Figura A.26 – Definição da relação VOLITIONAL CAUSE

Nome da relação: VOLITIONAL RESULT
Restrições sobre N: não há **Restrições sobre S**: apresenta uma ação volitiva ou uma situação que poderia surgir de uma ação volitiva **Restrições sobre N+S**: N apresenta uma situação que pode ter acarretado o fato do agente da ação volitiva em S ter realizado a ação; sem N, o leitor poderia não reconhecer a motivação da ação; N é mais central para a satisfação do objetivo do escritor do que S **Efeito**: o leitor reconhece a situação apresentada em N como a causa da ação apresentada em S

Figura A.27 – Definição da relação VOLITIONAL RESULT

Nome da relação: CONTRAST
Restrições sobre os Ns: não mais do que dois Ns; as situações nos Ns são (a) compreendidas como similares em vários aspectos, (b) compreendidas como diferentes em vários aspectos e (c) comparadas em relação a uma ou mais dessas diferenças **Efeito**: o leitor reconhece as similaridades e diferenças resultantes da comparação sendo feita

Figura A.28 – Definição da relação CONTRAST

Nome da relação: JOINT
Restrições sobre os Ns: não há **Efeito**: não há

Figura A.29 – Definição da relação JOINT

Nome da relação: LIST
Restrições sobre os Ns: itens comparáveis apresentados nos Ns **Efeito**: o leitor reconhece como comparáveis os itens apresentados

Figura A.30 – Definição da relação LIST

Nome da relação: SAME-UNIT
Restrições sobre os Ns: os Ns apresentam informações que, juntas, constituem uma única proposição **Efeito**: o leitor reconhece que as informações apresentadas constituem uma única proposição; separadas, não fazem sentido

Figura A.31 – Definição da relação SAME-UNIT

Nome da relação: SEQUENCE
Restrições sobre os Ns: as situações apresentadas nos Ns são realizadas em seqüência **Efeito**: o leitor reconhece a sucessão temporal dos eventos apresentados

Figura A.32 – Definição da relação SEQUENCE

Apêndice B – Relações Retóricas e seus Marcadores Textuais

Apresenta-se, neste apêndice, o conhecimento lingüístico resultante da análise do CorpusTCC.

A seguir, nas Tabelas D.1-D.27, são mostrados exemplos representativos dos marcadores textuais observados para cada relação retórica e os trechos de texto em que eles ocorrem, trechos estes extraídos do próprio corpus. Nas tabelas, mostram-se a ordem entre as proposições expressas no texto, isto é, satélite seguido por núcleo (SN), núcleo seguido por satélite (NS) ou núcleo seguido de outro núcleo (NN) (no caso de relações multinucleares), e exemplos, ou seja, as proposições conectadas pelas relações, com os possíveis marcadores textuais destacados em negrito. Quando o trecho de texto é muito grande e, por isso, somente uma parte sua é mostrada na tabela, o sinal de reticências (...) é utilizado para indicar isso. Note que não se diferenciam as relações volitivas das não volitivas para CAUSE e RESULT, dado que elas possuem os mesmos marcadores.

Tabela D.1 – Exemplos e marcadores superficiais para a relação ANTITHESIS

Ordem	Proposição 1	Proposição 2
NS	Com esse critério é possível testar a interface entre as unidades que compõem o software,	**ao contrário** da Análise de Mutantes, que explora as características das unidades separadamente.
	Com esse objetivo, muitas técnicas vêm sendo criadas visando à produção automática de sumários, dividindo-se principalmente em dois tipos de abordagens: profunda e superficial. A primeira utiliza teorias e modelos lingüísticos e técnicas de processamento de línguas naturais, enquanto a segunda utiliza técnicas estatísticas e empíricas, produzindo sumários pela extração de trechos do texto-fonte.	**No entanto**, como a tarefa de sumarizar textos é bastante complexa, apesar dos avanços da área, nenhuma das abordagens levou a um sumarizador automático que tenha alcançado resultados plenamente satisfatórios.
	Através desta definição, será possível o armazenamento de som no MRO, permitindo a recuperação do som desejado.	**É importante esclarecer** que, **independente** dos atributos criados, não será possível a recuperação de som através da sua reprodução.
SN	Após a modificação do programa, todo o conjunto de casos de teste, utilizado durante o teste original, deveria ser executado novamente para revelar possíveis defeitos introduzidos com a manutenção, comparando-se as saídas	**Apesar** dessa estratégia facilitar a geração de casos de teste para o teste de regressão, o seu uso pode tornar-se proibitivo devido ao tempo requerido para execução dos testes e análise dos resultados, onde todo o programa é testado

	obtidas pelo programa modificado com as saídas registradas pelo programa original.	novamente, até mesmo as partes do programa que não foram afetadas pelas modificações.
	Para defeitos de software existem poucas ferramentas; uma delas é a FINE, uma ferramenta mais abrangente que aborda tanto defeitos de hardware como de software.	**Em paralelo**, existem diversos trabalhos que buscam entender e classificar os defeitos que ocorrem em sistemas de software.
	Utilizando ferramentas de teste o conjunto de casos de teste pode ser facilmente obtido para a realização do teste de regressão.	**Entretanto**, é necessário que esse conjunto seja o mais adequado possível e, principalmente, mínimo para melhorar a relação custo-eficiência desta atividade.
	Os modelos são fechados, eles indicam o que deve ser feito,	**mas** não dizem como as atividades que proporcionam a melhoria devem ser realizadas.
	Os sistemas orientado a objetos caracterizam-se por suportar estruturas de dados bem mais elaboradas, além de permitir que a manipulação dessas estruturas seja também definida como parte delas próprias.	**No entanto**, dentro das técnicas usuais da disciplina de banco de dados, é de praxe suportar ao nível do gerenciador de banco de dados, as operações usuais sobre os tipos de dados, que permitem a otimização das consultas sobre esses tipos de dados comuns às aplicações suportadas.
	O objetivo inicial do trabalho é desenvolver um verificador ortográfico, uma ferramenta capaz de verificar se as palavras de um texto de entrada estão presentes ou não em um vocabulário da língua portuguesa. Caso não seja verificada uma ocorrência, o fato é notificado ao usuário para que ele possa optar pela ação desejada, como por exemplo, ignorar o fato, corrigir a palavra ou acrescenta-la ao seu dicionário pessoal. Para facilitar as correções, a ferramenta provê também um recurso de aconselhamento ortográfico, que propõe palavras do vocabulário com alguma semelhança com aquela não encontrada.	Pode-se perceber, **no entanto**, que verificar se uma palavra está presente ou não em um vocabulário não é suficiente para garantir que um texto foi editado corretamente.
	A flexibilidade e a facilidade de uso de hiperdocumentos na Web têm garantido um futuro cada vez mais promissor para a utilização de sistemas de hipertexto.	**Porém**, quando a construção de hiperdocumentos envolve a montagem de milhares de páginas e centenas de milhares de links, ela se torna uma atividade que pode gerar muitas informações inconsistentes.
	A comunidade de Tolerância a Defeitos no Brasil tem adotado o termo falha para fault, erro para error e defeito para failure, assim, referindo-se ao termo Fault Injection por Injeção de Falhas.	Martins **salienta que não** existe ainda um consenso quanto à terminologia a ser utilizada em Português para os termos fault, error e failure.

Tabela D.2 – Exemplos e marcadores superficiais para a relação ATTRIBUTION

Ordem	Proposição 1	Proposição 2
SN	Streitz **comentava que**	essa união resultaria em sistemas hipermídia multi-usuários distribuídos, ao passo que as atividades de cooperação seriam beneficiadas com o suporte a documentos estruturados.
	Maldonado **destaca que**,	reconhecido o caráter complementar das técnicas e critérios de teste e a diversidade de critérios que têm sido estabelecidos, um ponto crucial que se coloca nessa perspectiva é a escolha e/ou a determinação de uma estratégia de teste, que em última análise passa pela escolha de critérios de teste, de forma que as vantagens de cada um desses critérios sejam combinadas objetivando uma atividade de teste de maior qualidade.
	Borko **dizia que**	o processamento automático da língua nunca seria perfeito, mas que poderia ser de alta qualidade.
	Nesse mesmo contexto, Ishii **relatava que**	a tecnologia de hipertexto deveria beneficiar o suporte computacional ao trabalho em grupo,
	Maldonado **ressalta que**,	do ponto de vista de qualidade do processo, o teste sistemático é uma atividade fundamental para a ascensão ao Nível 3 do Modelo CMM 2 do SEI.
	Além da Injeção de Defeitos, **Geist sugere que**	o critério Análise de Mutantes poderia ser valioso no projeto (desenvolvimento de software) de sistemas de software tolerantes a defeitos.
	Desenvolver sistemas dessa forma tem seus problemas**:**	na ânsia de aproveitar algo já pronto, muitas vezes o sistema resultante não fica tão eficiente
	Existem três etiquetadores para o Português contemporâneo do Brasil **-**	o etiquetador estatístico desenvolvido na UFRGS, apresentando uma precisão de 84,5%, o etiquetador neural desenvolvido na Universidade Nova de Lisboa, com a precisão 88.7%, e o etiquetador baseado em regras, desenvolvido por Eckhard Bick, que possui uma precisão acima de 99%

Tabela D.3 – Exemplos e marcadores superficiais para a relação BACKGROUND

Ordem	Proposição 1	Proposição 2
SN	**Este projeto é baseado** em um SGBDOO – Sistema Gerenciador de Banco de Dados Orientado a Objetos – chamado GEO – GErenciador de Objetos. O GEO tem sido construído no Instituto de Ciências Matemáticas de São Carlos com o objetivo de validação prática e estudo de como os	**A proposta deste trabalho** é a inclusão no MRO (e implementação do respectivo suporte no GEO) de uma nova característica: áudio. O objetivo dessa característica é permitir a manipulação de sons como música e voz, entre outros, e sua definição deverá levar em conta essas

	conceitos de orientação a objetos podem ser implementados em um sistema de software real. Este gerenciador é baseado no modelo de dados denominado Modelo de Representação de Objetos – MRO.	modalidades de áudio.
	Na última década houve um crescimento explosivo no uso de dados e aplicações multimídia. **Hoje em dia**, computadores e redes processam e transmitem muito mais que apenas texto e imagens estáticas. Mídias contínuas, como vídeo e áudio, juntamente com mídias discretas, como gráficos, se tornaram parte de aplicações integradas de computador.	**As novas aplicações que estão surgindo** fazem crescer a cada dia a necessidade de distribuição de dados e de processamento. Entre essas novas aplicações estão os grandes sistemas hipermídia, que são inerentemente distribuídos e por isso requerem a adoção de padrões para representação e intercâmbio de dados multimídia.
	Os últimos anos têm apresentado um grande aumento na aceitação e adoção do processamento paralelo, tanto para computação científica de alto desempenho como para aplicações de propósito geral. Essa aceitação tem sido favorecida principalmente pelo desenvolvimento dos ambientes com processamento maciçamente paralelo. (...)	**Assim, o objetivo deste trabalho** é verificar as afirmações discutidas nos parágrafos anteriores, ou seja, segundo Geist o MPI é mais adequado para arquiteturas paralelas, enquanto o PVM se adapta melhor em sistemas distribuídos.
	Problemas com a escrita podem afetar o desempenho de profissionais de maneira marcante, principalmente no caso de pesquisadores e acadêmicos que precisam escrever com proficiência e desembaraço não apenas na língua materna, mas também em uma ou mais línguas estrangeiras.	**Atualmente**, o inglês é a língua dominante para a escrita e divulgação de pesquisas científicas na forma de artigos científicos.
	O modelo de referência Dexter, mesmo apresentando arquitetura genérica para sistemas hipermídia, não cobre adequadamente todos os aspectos que distinguem sistemas hipermídia abertos de outros sistemas hipermídia. Segundo Osterbye e Wiil, um aspecto não definido em Dexter é a possibilidade de fácil integração e uso de ferramentas viewer para editar e mostrar componentes hipermídia. No trabalho de Osterbye e Wiil é apresentado um framework denominado Flag para sistemas hipermídia abertos construído sobre a arquitetura inicialmente proposta no modelo Dexter. (...)	**Com base** nas características relevantes para a especificação de aplicações em sistemas hipermídia abertos presentes no framework Flag proposto por Osterbye e Wiil e nos requisitos destacados por Paulo, **um objetivo inicial desta pesquisa** se concentra em uma análise sobre a adequação de técnicas formais existentes, como Redes de Petri, Statecharts e CSP (Communicating Sequential Process), em relação a especificação de aplicações hipermídia abertas.
	Visando a facilitar a especificação do aspecto comportamental de Sistemas Reativos, várias técnicas de especificação foram propostas. Essas técnicas tentam conciliar o poder de modelagem com a capacidade de análise de propriedades do sistema. Dentre essas técnicas estão as baseadas em modelos de Máquinas de	**Considerando essas técnicas**, Fabbri explorou a adequação do uso do conceito de mutação no contexto de teste e validação de aspectos comportamentais de Sistemas Reativos. O critério Análise de Mutantes e critérios alternativos denominados Mutação Restrita e Mutação Aleatória, bem como os operadores de

	Transição de Estados, nas quais o aspecto comportamental é modelado através de eventos discretos no tempo. Como exemplos de técnicas dessa categoria, têm-se: Máquinas de Estados Finitos, Statecharts e Redes de Petri.	mutação para essas técnicas também foram definidos. Para apoiar a aplicação da Análise de Mutantes neste contexto, Fabbri especificou a ferramenta Proteum-RS (PROduct TEsting Using Mutation for Reactive Systems), e a instanciou para apoiar a validação de especificações baseadas em Máquinas de Estados Finitos, originando a Proteum-RS/FSM. (...)
	A crescente popularização dos computadores, trazida pelas praticidades e comodidades que eles proporcionam, gera esforços sucessivos em aprimoramentos, tanto em hardware quanto em software, para que a interação homem-máquina seja cada vez mais descomplicada e transparente. (...)	**E é justamente neste contexto** que se insere nosso trabalho. **O objetivo** é fazer um estudo comparativo do uso das técnicas conexionista e simbolista, na revisão automática de erros gramaticais da língua portuguesa. (...)
	Uma ferramenta que apoie a fase de engenharia de requisitos deve considerar inicialmente as especificações geradas por intermédio da análise das necessidades do sistema após consultas aos usuários. Existem várias maneiras para obter e analisar requisitos, uma técnica já conhecida que tem obtido grande destaque atualmente é a que utiliza a técnica de criação de cenários, que permitem retratar, a partir das necessidades dos usuários, os caminhos possíveis para utilização do sistema.	**Este trabalho enfoca** principalmente o uso de cenários na engenharia de requisitos e na construção de uma ferramenta para apoio a esse uso.
	A Engenharia de Software é uma disciplina em evolução e que está em consonância com a tecnologia de computadores e com os requisitos de novas áreas de aplicação, tendo como objetivo principal produzir softwares de alta qualidade e de baixo custo. Abrange inúmeras áreas de pesquisa: engenharia de requisitos, projeto, verificação, validação e teste, manutenção, planejamento, gerenciamento de configuração, entre outras. (...)	**Este trabalho propõe** um esquema de injeção de defeitos de software baseado na taxonomia de defeitos de DeMillo e nos operadores de mutação do critério de teste Análise de Mutantes para a geração e injeção de defeitos.
	Nas últimas décadas, pode-se considerar que os critérios baseados em Análise de Fluxo de Dados (técnica estrutural) e o critério Análise de Mutantes (técnica baseada em erros) constituem as contribuições mais relevantes na área de teste. (...)	**Este trabalho segue esta perspectiva**, contribuindo para a redução do custo de aplicação do critério Análise de Mutantes na medida em que fornece subsídios para a determinação de um subconjunto do total de operadores de mutação (conjunto essencial), facilitando, com isso, a condução da atividade de teste.
	Nas aplicações hipermídia, o aprendiz explora documentos organizados por páginas (ou nós) e estruturados através de ligações (ou elos), com uma rica variedade de associações entre as informações. O	**Nesse contexto** de ensino, o controle que o aprendiz é capaz de ter sobre o material didático em aplicações hipermídia é uma questão importante por causa da responsabilidade que este passa a ter no

	sistema apenas fornece o material e proporciona uma forma de navegação através dele, com o controle da interação totalmente a cargo do aprendiz, permitindo que este tenha progresso de acordo com os seus interesses e objetivos. (...)	processo de aprendizagem. Ele deve decidir que caminhos seguir, quando voltar ou pular adiante, quando seguir um determinado caminho e quando evitar as distrações com possíveis caminhos irrelevantes. (...)
	Após o surgimento de técnicas e métodos sistemáticos, especialmente elaborados para apoiar o desenvolvimento de software, várias alterações e melhorias foram propostas para essa atividade, o que causou uma grande revolução na maneira segundo a qual o software era criado. As primeiras implementações, resultantes do surgimento do elemento software, em contrapartida ao elemento hardware, eram realizadas sem qualquer tipo de administração, o que resultava, na maioria das vezes, em prazos esgotados e em custos elevados. (...)	**Nesse contexto**, esse projeto de pesquisa propõe a validação do Educational Hyperdocuments Design Tool (EHDT), especialmente desenvolvido para possibilitar a modelagem de hiperdocumentos para o Sistema de Autoria e Suporte Hipermídia para Ensino (SASHE), e que foi implementado segundo os conceitos do Educational Hyperdocuments Design Method (EHDM). (...)
	A Engenharia de Software é tipicamente uma das áreas da Ciência de Computação que envolve, além de um grande volume de documentos, uma grande diversidade de tipos de documentos (diagramas, textos, códigos-fonte, executáveis, etc.). Tal característica, aliada ao fato de que tais documentos contêm informações bastante relacionadas, sugere naturalmente a utilização de hiperdocumentos (com nós e links) como meio adequado para o armazenamento e recuperação dessas informações. (...)	**Neste trabalho de mestrado**, o principal interesse foi o tratamento de hiperdocumentos para dar o suporte necessário ao processo de engenharia reversa de software. Dessa forma, os estudos neste trabalho tiveram como alvo a construção de um hiperdocumento que possibilitasse a fidelidade do conteúdo da documentação com relação ao produto de software sendo documentado. Além disso, tivemos como meta obter a consistência entre as partes do hiperdocumento e os componentes do software com mais facilidade por meio dos links definidos. (...)
	Apesar de ser uma técnica conhecida há algum tempo, os cenários têm ganhado nos últimos anos grande destaque entre os principais autores na área de desenvolvimento de sistemas. Há vários métodos publicados recentemente, entre eles OMT, Objectory e Fusion, que utilizam a técnica de construção de cenários em suas fases e uma grande quantidade de extensões que utilizam cenários como técnica de apoio.	**Neste trabalho serão estudadas** as fases de engenharia de requisitos de alguns métodos que utilizam a técnica de construção de cenários, com o objetivo de avaliar os pontos críticos que devem compor uma técnica completa para elicitar requisitos usando cenários e desenvolver uma ferramenta, baseada nessa técnica, que apoie a fase de engenharia de requisitos.
	Segundo Pressman, existem diversas categorias de software: software básico, sistemas de informação, sistemas científicos, sistemas embutidos, sistemas pessoais, sistemas de inteligência artificial e sistemas reativos. Um Sistema Reativo é um programa de computador que mantém um interação permanente com seu ambiente externo, o qual pode ser um	**O contexto mais geral deste trabalho** é a investigação de estratégias de teste e validação para o desenvolvimento de Sistemas Reativos, com ênfase no critério Análise de Mutantes.

	usuário, um dispositivo de entrada ou uma outra parte do sistema. (...)	
	Sistemas baseados em computação estão sendo utilizados em praticamente todas as áreas da atividade humana, provocando uma crescente demanda por qualidade e produtividade. A Engenharia de Software é uma disciplina que evoluiu nas últimas décadas procurando estabelecer técnicas, critérios, métodos e ferramentas para a produção de software.	**Sendo assim, este trabalho contribui para** o estabelecimento de estratégias de teste e validação para o desenvolvimento de Sistemas Reativos/Concorrentes, com ênfase no critério Análise de Mutantes e na técnica Statecharts, dando continuidade ao trabalho desenvolvido por Fabbri.

Tabela D.4 – Exemplos e marcadores superficiais para a relação CAUSE

Ordem	Proposição 1	Proposição 2
NS	O PVM é considerado um padrão de fato,	**dado** a sua vasta utilização e popularidade alcançadas não apenas no meio acadêmico, mas também em diversas aplicações na indústria e outros setores.
	A construção de arquiteturas dedicadas tem sido uma abordagem pouco explorada,	**devido ao** elevado custo de projeto e tempo de desenvolvimento.
	Em adição, os desenvolvedores de páginas Web encontram dificuldades quando muitas pessoas estão envolvidas na construção em paralelo de uma mesma página ou de um conjunto de páginas relacionadas.	**Isso se deve ao fato** de que os desenvolvedores trabalham independentemente em suas próprias cópias, tendo como principal problema a integração dessas cópias em um hiperdocumento final.
	Sempre haverá casos que não serão tratados, mesmo que tenhamos um lingüista genial para elaborar regras ou um etiquetador perfeito,	**isto porque** não é possível construir um corpus que inclua todas as enunciações de uma língua dada ou subconjunto de uma língua, exceto para algumas línguas mortas, em que a quantidade de textos disponíveis é limitada.
	contribuindo para a redução do custo de aplicação do critério Análise de Mutantes	**na medida em que** fornece subsídios para a determinação de um subconjunto do total de operadores de mutação (conjunto essencial), facilitando, com isso, a condução da atividade de teste.
	Contudo, a construção de aplicações distribuídas impõe novos desafios aos programadores,	**pela** diversidade dos ambientes computacionais envolvidos, pela necessidade de troca de mensagens através da rede e devido à própria compreensão do que vem a ser uma aplicação distribuída.
	Entretanto, eles possuem diálogo limitado,	**pois** estão presos à sua estrutura rígida de interação.
	Nesse contexto de ensino, o controle que o aprendiz é capaz de ter sobre o material didático em aplicações hipermídia é uma questão importante	**por causa da** responsabilidade que este passa a ter no processo de aprendizagem. (...)
	A verificação do software através de modelagem analítica torna-se complicada	**por** depender de um grande número de variáveis.
	A demanda cognitiva sobre o aluno é maior	**porque** ele deve tomar as decisões corretas para que o aprendizado ocorra de

		fato.
	Mas ainda existem 50% dos 10% restantes que fazem com que a etiquetagem não seja uma tarefa trivial.	**Estes 50% são resultado**, por exemplo, dos problemas: - Ambigüidade léxica que não pode ser resolvida pelo contexto - existe a ambigüidade mas o contexto em que as palavras ambíguas aparecem é o mesmo. (...)
	Atualmente, no entanto, a comunidade de PLN do Brasil vive um novo período de entusiasmo e resultados promissores.	**Um dos motivos para isso** é a possibilidade de contar com ferramentas de tratamento lingüístico bastante abrangentes e independentes de língua, que facilitam sobremaneira a implementação de aplicativos nessa área.
	Alguns autores consideram sistemas CSCL como uma subdivisão dos sistemas CSCW dedicados às aplicações educacionais,	**uma vez que**, muitas vezes, suportam algumas atividades básicas do trabalho cooperativo ao mesmo tempo em que agregam elementos associados a atividades de aprendizagem e tutoria.
SN	**e devido ao** tamanho do código fonte e da complexidade,	**acabam por** dificultar a sua manutenção.
	Definida como uma aplicação de SGML, HTML (Hypertext MarkUp Language) é uma linguagem que fez sucesso por sua simplicidade. Entretanto, uma única linguagem não consegue suportar de modo satisfatório as inúmeras aplicações existentes hoje na Web.	**Assim**, ênfase tem sido dada à definição de padrões, recomendações e formatos para a definição, manipulação e intercâmbio de hiperdocumentos estruturados suportados por sistemas hipermídia.
	Isso torna o sistema independente de um vocabulário específico,	**aumentando** a portabilidade e restringindo o problema causado pela presença de palavras desconhecidas.
	Essas técnicas se diferenciam, basicamente, pela origem da informação usada para avaliar ou para construir conjuntos de casos de teste, onde cada técnica possui um conjunto de critérios para esse fim.	**Com isso**, nenhuma das técnicas de teste é suficiente isoladamente para garantir a qualidade da atividade de teste.
	Como as organizações solicitam as mudanças de forma muito aleatória, todas as mudanças feitas devem ser registradas e efetuadas no sistema a um custo razoável,	**daí surge** a necessidade delas serem gerenciadas.
	Segundo Pressman, quanto mais tarde um erro for encontrado no processo de desenvolvimento de software, maior é o custo para correção desse erro.	**Dessa forma**, o ideal é a detecção de erros no início do processo, o que motiva o uso de critérios sistemáticos para o teste e validação do sistema ainda na fase de especificação.
	Segundo Garzotto, o maior desafio encontrado pelos autores de aplicações hipermídia é capturar e organizar assuntos complexos de maneira adequada, facilitando posteriormente a sua manutenção.	**Desse modo**, uma abordagem sistemática para modelar o conteúdo e definir a organização estrutural é especialmente importante no projeto de aplicações hipermídia grandes e complexas. (...)
	Tendo em vista que, em geral, a ocorrência de falhas em Sistemas Reativos pode colocar em risco vidas humanas e/ou	**a atividade de teste nesse contexto deve ser** realizada com rigor, mesmo quando isso implique em altos custos.

	levar a grandes prejuízos materiais,	
	Como as organizações solicitam as mudanças de forma muito aleatória,	**todas as mudanças feitas devem ser** registradas e efetuadas no sistema a um custo razoável,
	Uma vez que os Sistemas Reativos distinguemse de outros tipos de sistemas por terem no comportamento sua maior ênfase,	**devem-se** buscar formas de teste e validação do aspecto comportamental das especificações de tais sistemas.
	Essas soluções poderiam ser sintetizadas em padrões, sejam eles de análise, de projeto ou de código. Para criar esses padrões nada mais natural do que investigar sistemas prontos, em busca de trechos de código que representem soluções para determinados problemas e que possam ser reutilizados no futuro.	**Deve-se estudar, então,** uma forma de documentar esses padrões, disponibilizando-os a quem possa interessar.
	Entretanto, tais modelos ainda estão no seu início, sendo que muitos deles não são eficientes e expressivos,	**dificultando** a sua utilização.
	Devido a crescente importância que as redes de computadores tem adquirido,	é essencial uma ferramenta de simulação de redes de computadores para o ASiA.
	que implementa os processos de apoio à cooperação,	**e assim possibilita** o trabalho conjunto, bem como a necessária troca de informações
	Como a UML é uma notação padrão para os modelos orientados a objetos, independente do processo de desenvolvimento a ser seguido,	**é necessário** ou adaptar os processos de desenvolvimento de software existentes ou desenvolver novos processos que suportem a notação proposta pela UML.
	Portanto, os Sistemas Reativos controlam algumas atividades humanas essenciais	**e por isso**, a atividade de teste no desenvolvimento dos mesmos é ainda mais crucial
	Sistemas baseados em computação têm sido utilizados em todas as áreas da atividade humana	**e, como conseqüência**, aspectos de qualidade e produtividade somam-se à inerente dificuldade e complexidade da atividade de desenvolvimento de software.
	devido à ampla abrangência desse tema,	este projeto se restringirá a definir com detalhes apenas a representação de sons musicais.
	O avanço tecnológico verificado nos últimos anos resultou na redução de custo e no aumento do poder computacional, na proliferação de software para processamento de dados geo-referenciados, na disponibilidade de controladores de tempo real e de sistemas de navegação de precisão e no desenvolvimento de sensores eletrônicos.	**Estes fatores tornaram** mais fácil e confiável a aquisição de dados em tempo real necessária na atividade de sensoriamento remoto.
	A maioria das abordagens de teste orientado a objetos é baseada em implementação, não sendo propostas técnicas de teste OO baseadas em especificação.	**Existe então** a necessidade de desenvolver abordagens que permitam a geração de conjuntos de teste baseados na especificação do software.
	e por serem formados por pequenas partes	**facilitam** a manutenção.
	na medida em que fornece subsídios para a	**facilitando, com isso**, a condução da

	determinação de um subconjunto do total de operadores de mutação (conjunto essencial),	atividade de teste.
	e atualmente está tornando o seu navegador como o núcleo de seu sistema operacional,	**fazendo com que** o conteúdo de todo winchester seja um hiperdocumento.
	Da necessidade de trabalhar com as RNCs em um único ambiente,	o Simulador Kipu foi inicialmente projetado.
	Procurar-se-á complementar e comparar os experimentos realizados por Wong et al para avaliar ambas as técnicas de teste de regressão,	**gerando** conhecimento e experiência na perspectiva do estabelecimento de estratégias de revalidação eficazes e de baixo custo.
	Nesse caso, a transmissão muitas vezes fica prejudicada devido à estrutura das redes intermediárias.	**Logo**, a demora para recuperar as informações pode quebrar o fio de pensamento do aprendiz.
	O controle que o estudante tem possibilita-o a realizar escolhas afetando o andamento do aprendizado,	**o que resulta** num estudante se sentindo mais competente, bem como a atividade de estudar se torna mais pessoal e interessante.
	As palavras do vocabulário do ispell são armazenadas numa tabela hash,	**o que torna** a pesquisa eficiente.
	A reengenharia com mudança de linguagem é feita tanto de forma automática como de forma manual,	**obtendo-se** sistemas em linguagens orientadas a objetos.
	Quando se trata de textos, o conteúdo dos nós é, na maioria das vezes, uma informação resultante de processos de escrita e formatação.	**Por esse motivo**, os sistemas para verificação de ortografia, corretores gramaticais e sistemas de processamento de linguagem natural são exemplos de ferramentas apropriadas à avaliação automática e auxílio à autoria neste nível de granularidade fina de informações contidas no hipertexto.
	As páginas são freqüentemente marcadas como em construção sem nenhuma informação sobre quando a construção começou ou quando vai terminar.	**Por isso** os leitores têm que revisitar as páginas para verificar se elas já estão completas e disponíveis.
	Barros identifica a cooperação como um fenômeno que envolve vários processos: comunicação, negociação, coordenação, co-realização e compartilhamento.	**Portanto**, para que as pessoas trabalhem cooperativamente, em um mesmo local ou geograficamente distribuídas, é necessário que exista entre elas um ambiente de apoio à comunicação.
	Sistemas baseados em computação estão sendo utilizados em praticamente todas as áreas da atividade humana,	**provocando** uma crescente demanda por qualidade e produtividade.
	Há uma grande quantidade de informações circulando na Internet provenientes de várias partes do mundo e novas áreas de aplicação que necessitam de uma rica variedade de tipos de dados multimídia estão emergindo.	**Tal acontecimento fez com que** os sistemas hipermídia se tornassem o mais recente meio de comunicação mundial, além de aumentar o desenvolvimento, em escala comercial, das aplicações hipermídia. (...)
	Geralmente as empresas de pequeno ou médio porte não possuem uma ampla infra-estrutura, um grande número de pessoas envolvidas no processo e nem tampouco	**tornando** praticamente inviável a aplicação de modelos de melhoria tais como o SW-CMM.

	recursos e tempo disponíveis,	
	Por ser um modelo novo,	a notação gráfica do SIRIUS não foi validada através da utilização em aplicações práticas.
	Uma vez que a conclusão do novo projeto consumiu mais dedicação do que inicialmente previsto,	apenas os primeiros experimentos planejados foram executados. (...)

Tabela D.5 – Exemplos e marcadores superficiais para a relação CIRCUMSTANCE

Ordem	Proposição 1	Proposição 2
NS	pois o usuário (leitor/aprendiz) tende a perder o sentido de localização e direção das informações (desorientação)	**à medida que** navega de uma página para outra em uma estrutura rasa, ou não-hierárquica, de relacionamentos.
	Essa visão é baseada na opinião de que as decisões sistemáticas e a estrutura racional da aplicação devem ser tomadas	**antes** de ela ser implementada (...)
	onde o estudante não se sinta intimidado	**ao cometer** erros.
	A manutenção envolve qualquer mudança feita	**após** o software estar em uso.
	Durante seu treinamento, uma RNC insere novos neurônios e conexões de acordo com um critério definido por seu Algoritmo Construtivo,	**até que** uma solução satisfatória seja encontrada ou o algoritmo seja interrompido.
	Durante esse processo, pode-se fazer uma busca por situações	**em que** esses padrões encontrados possam ser empregados (...)
	que auxiliará o estudante,	**enquanto** ele estiver navegando em um hiperdocumento (...)
	Os sistemas hipermídia respondem estas perguntas	**na hora em que** elas são geradas (...)
	Desse modo, a determinação de um conjunto essencial de operadores de mutação para a linguagem C,	**na medida em que** reduz o custo de aplicação do critério Análise de Mutantes (...)
	Esse sistema é muito utilizado em situações	**onde** a informação evolui naturalmente através de uma série de versões, onde existe a necessidade de acessar versões anteriores da mesma forma que a versão atual e quando a informação é produzida com um esforço colaborativo por equipes de pessoas responsáveis pela criação da informação.
	Essas aeronaves são conhecidas por aeromodelos,	**principalmente quando** são utilizadas para entretenimento.
	Mesmo com mecanismos poderosos de armazenamento e transmissão, não é uma boa idéia enviar um vídeo completo ao cliente	**quando** se deseja saber apenas o assunto sobre o qual o vídeo trata.
	ao contrário da maioria das estratégias existentes que apóiam o teste	**somente depois** que o software foi programado.
SN	O fato de se ter um programa testado não garante a ausência de defeitos, o que não permite afirmar que o programa está correto.	**Assim**, o teste de software tem o objetivo de revelar defeitos, contribuindo para demonstrar que as funções do software estão sendo desempenhadas de acordo com a especificação.

	Uma das linhas de atuação do Grupo de Engenharia de Software do ICMC/USP (em conjunto com o Grupo de Teste do DCA/FEE/UNICAMP) tem concentrado suas atividades no estudo de princípios, estratégias, métodos e critérios de teste e validação na produção de software, bem como na especificação e implementação de ferramentas que apóiem a realização da atividade de teste e viabilizem a avaliação do aspecto complementar dos critérios, através de estudos empíricos.	**Dentro desse contexto**, três ferramentas de teste foram desenvolvidas -- a Poke-Tool, a Proteum e a Proteum/IM -- que apóiam os critérios Potenciais-Usos, Análise de Mutantes e Mutação de Interface (Interface Mutation), respectivamente (...)
	Dessa forma, considerando-se a concentração de trabalhos relacionados ao desenvolvimento de sistemas hipermídia e as limitações observadas nos trabalhos revisados relacionados à definição de requisitos para o desenvolvimento desses sistemas,	**este trabalho propõe** um conjunto único de requisitos, que possui as propriedades de ser abrangente e suficientemente completo para auxiliar a etapa de engenharia de requisitos de um novo sistema ou para permitir a avaliação de um sistema já existente.
	Considerando os conceitos apresentados acima, como a importância das atividades de teste de software durante o processo de desenvolvimento de software, a importância das atividades do teste de regressão durante a evolução/manutenção do software e a necessidade de minimização dos custos dessas atividades,	**este trabalho tem por objetivo** realizar estudos empíricos e comparativos utilizando as técnicas de teste de regressão propostas por Wong et al: Técnica baseada em Modificação e Técnica baseada em Mutação Seletiva.
	Entre os estudos empíricos que visam a estabelecer alternativas viáveis para a aplicação do critério Análise de Mutantes pode-se destacar o trabalho de Offutt et al: a partir dos operadores da ferramenta de teste Mothra, que apóia a aplicação da Análise de Mutantes para programas escritos em Fortran, Offutt et al determinaram um conjunto essencial de operadores de mutação para esta linguagem. (...)	**Na mesma perspectiva** dos estudos de Offutt et al e Wong et al, este trabalho tem como objetivo investigar alternativas pragmáticas para a aplicação do critério Análise de Mutantes e, nesse contexto, é proposto um procedimento para a determinação de um conjunto essencial de operadores de mutação para a linguagem C, com base nos operadores implementados na ferramenta Proteum.
	Para que a atividade de injeção de defeitos seja realizada de forma mais objetiva e eficaz, são requeridos modelos de defeitos e métodos de injeção de defeitos. (...)	**Nesse contexto**, diversas ferramentas têm sido projetadas e implementadas. (...)
	Para o caso de arquiteturas de computadores, a simulação é especialmente atrativa, principalmente nos casos em que diversas arquiteturas ou diferentes mecanismos para melhorar o seu desempenho são considerados. (...)	**Neste sentido**, vem sendo desenvolvido, pelo Grupo de Sistemas Distribuídos e Programação Concorrente do ICMSC-USP, o ASiA (Ambiente de Simulação Automático) com o objetivo de automatizar o processo de construção do programa de simulação.
	Como já citado, diferentes algoritmos têm sido empregados na fusão de sensores.	**Neste trabalho** optou-se por investigar diferentes técnicas de Inteligência Artificial como algoritmos para fusão. (...)
	Tomando por base o sistema Hip/Windows previamente construído,	**o objetivo deste trabalho é** o de complementar o ambiente de ensino

		(estudante) com funções diretas, não inseridas dentro das ligações (links), procurando: aprimorar a interação com o estudante, facilitar em situações de dificuldade ou dúvida, ajudar na procura de informações complementares, etc.
	Pensando um pouco num estágio posterior,	**procura-se** também oferecer ao professor recursos que facilitem a análise das informações criadas durante a navegação do estudante.
	Com base nas características relevantes para a especificação de aplicações em sistemas hipermídia abertos presentes no framework Flag proposto por Osterbye e Wiil e nos requisitos destacados por Paulo,	**um objetivo inicial desta pesquisa** se concentra em uma análise sobre a adequação de técnicas formais existentes (...)
	à medida que o software evolui,	o conjunto de casos de teste aumenta e, conseqüentemente o custo do teste de regressão.
	A partir dessa classificação dos sistemas computacionais voltados para o ensino,	torna-se mais fácil a identificação de características educacionais em sistemas computacionais, mais claras as vantagens e desvantagens proporcionadas por esses sistemas e o direcionamento de estudos e pesquisas sobre os recursos oferecidos por eles.
	Além disso, ao aceitar qualquer domínio,	esses métodos tendem a utilizar modelos de representação que são estranhos aos autores (...)
	Além disso, quando o sistema sofre alguma alteração,	é necessário refazer todas as equações e resolvê-las novamente.
	Antes de se definir os passos para melhorar o processo de software,	primeiramente as empresas devem realizar a avaliação desse processo.
	Ao criar uma seqüência de contextos contendo diferentes partes de uma seção,	cada parte possuirá um grau de liberdade.
	Assumindo que N é o número de caracteres da palavra em questão e M é o número médio de caminhos emergentes dos nodos,	o número de comparações feitas para uma palavra é M\Theta N.
	Com a popularidade da WEB e o aumento da demanda pela utilização de aplicações hipermídia que se integrem de forma natural a diferentes ambientes,	nota-se a necessidade de modelos voltados à especificação dessas aplicações fornecendo suporte, principalmente, a interoperabilidade.
	Em particular, em sistemas que suportam meta-estruturas, tal como é o caso do GEO,	essas operações podem ser feitas tanto no esquema de dados quanto no banco de dados propriamente dita.
	em se tratando de software paralelo,	o seu desenvolvimento está intimamente ligado à arquitetura alvo.
	Enfocando-se, por exemplo, os processadores de textos,	percebemos a grande evolução que sofreram e ainda sofrem no sentido da incorporação de ferramentas de auxílio à confecção de documentos.
	Na tentativa de solucionar essa situação,	apareceram os sistemas distribuídos, que tiveram como objetivo inicial o compartilhamento de recursos.

	Percebendo a importância da internet para os usuários,	ela tem inserido recursos voltados para a grande rede em todos os seus produtos e atualmente está tornando o seu navegador como o núcleo de seu sistema operacional, fazendo com que o conteúdo de todo winchester seja um hiperdocumento.
	Quando o estudante já possui algum conhecimento do problema a ser simulado,	ele não encontra problemas em identificar os recursos e as características que o sistema tem em comum com o mundo real.
	Seguindo o modelo de referência de Dexter,	um sistema hipertexto é dividido em três camadas: camada de tempo-de-execução (run-time), de armazenamento (storage) e interna-aos-componentes (within-component).

Tabela D.6 – Exemplos e marcadores superficiais para a relação COMPARISON

Ordem	Proposição 1	Proposição 2
NS	ao invés do material didático,	**como** acontecia normalmente.
	e responde	**como se** fosse o sistema sendo simulado (...)
	desta forma, a definição de um esquema conceitual ajuda a oferecer uma visão **mais** abstrata do domínio da aplicação	**do que** aquela obtida pela inspeção do seu código, ou pela tentativa de extrair uma semântica das estruturas dos nós e dos elos (...)
	permitindo uma flexibilidade no projeto de modelos e algoritmos de Redes Neurais ainda **maior**	**que** a encontrada na versão anterior do Simulador.
	mas particularmente pelo fato de suportarem um modelo de aprendizagem	**que contrasta** com o programa tradicional de ensino e com o modelo de sistemas tutores.
	Uma versão de um sistema é uma instância	**que difere**, de algum modo, de outras instâncias.
	as redes neurais seriam implementadas exatamente como são propostas nos modelos originais,	**sendo exatamente** o que dispomos no simulador.
SN	Dentro desta visão, **quanto mais** controle o usuário puder ter,	**maior** chance de sucesso haverá na interação com o sistema.

Tabela D.7 – Exemplos e marcadores superficiais para a relação CONCESSION

Ordem	Proposição 1	Proposição 2
NS	Durante algum tempo, o entusiasmo pela área de PLN esmoreceu,	**ainda que** vários resultados interessantes e promissores tenham sido alcançados.
	É preciso também desenvolver estratégias de teste de software que apóiem todas as fases do desenvolvimento do software,	**ao contrário** da maioria das estratégias existentes que apóiam o teste somente depois que o software foi programado.
	O estudante se torna o foco dos sistemas de instrução,	**ao invés** do material didático, como acontecia normalmente.
	Um fato importante é que alguns sistemas CSCW são utilizados no processo de aprendizagem,	**apesar de** não terem sido construídos para este propósito.
	Será utilizado também o termo Tolerância a	**embora** usualmente este seja empregado

	Defeitos referindo-se a Fault Tolerance,	como Tolerância a Falhas.
	As aplicações não convencionais, em geral aproveitam os SGBDs já existentes,	**mesmo que** estes não sejam adequados para estas aplicações.
SN	Desta forma, **por mais que** as arquiteturas tenham apresentado um processo evolutivo intenso,	**ainda** é necessário buscar um melhor desempenho para a execução das aplicações que crescem tanto em volume como em complexidade muito mais rapidamente que o hardware disponível.
	em vez de se procurar determinar a categoria sintática a partir de características da palavra ou do contexto frasal,	**apenas** listava-se a categoria de cada palavra em um léxico, junto com suas outras características.
	Dentre as técnicas de verificação e validação, o teste é, sem dúvida, a atividade mais utilizada. Segundo Myers, teste é um procedimento de executar um programa com a intenção de encontrar erros existentes. (...)	**Apesar** dos esforços durante a atividade de teste, não se pode garantir um software livre de erros.
	Os Sistemas Computacionais Distribuídos, há mais de uma década, já deixaram de ser apenas uma promessa, não estando mais restritos a pesquisas realizadas nos meios acadêmicos. (...)	**Contudo**, a construção de aplicações distribuídas impõe novos desafios aos programadores, pela diversidade dos ambientes computacionais envolvidos, pela necessidade de troca de mensagens através da rede e devido à própria compreensão do que vem a ser uma aplicação distribuída.
	não importa se para as diferentes organizações as pessoas são clientes, empregados, donos ou pacientes,	o significado de pessoas é sempre entendido pelos membros das organizações e muitos dos dados mantidos para pessoas em ambas as organizações têm o mesmo significado.
	Da necessidade de trabalhar com as RNCs em um único ambiente, o Simulador Kipu foi inicialmente projetado. Este simulador oferece recursos gráficos para a edição de Redes Neurais e permite a inclusão de novos modelos e algoritmos de treinamento.	**Entretanto**, no decorrer deste trabalho observou-se que o projeto original do Simulador Kipu não atenderia às necessidades impostas pela proposta de trabalho.
	Vários trabalhos de pesquisa têm mostrado esforços na tentativa de se formalizar uma gramática para o Português do Brasil, que seja passível de implementação visando a geração automática de sentenças ou parágrafos. (...)	**Infelizmente, no entanto,** nenhum trabalho científico nacional chegou a um termo tal que uma gramática (total ou parcial) do português do Brasil tenha sido de fato implementada de maneira efetiva, em qualquer dos formalismos existentes.
	O usuário tem a impressão de que tudo está acontecendo no seu equipamento,	**mas** na realidade ele está compartilhando discos, impressoras, processadores e tudo mais que for necessário e o sistema permitir.
	A estratégia de ensino utilizada aqui é única: estímulo e resposta.	**Mesmo assim**, torna-se eficaz para se alcançar treinamento, que é o objetivo principal desse tipo de sistema.
	Como dito anteriormente, o processo de desenvolvimento de software consiste de uma série de atividades,	sendo que, **mesmo com** o uso de métodos, técnicas e ferramentas, erros podem ser introduzidos no produto.
	O objetivo dessa característica é permitir a manipulação de sons como música e voz,	**No entanto**, devido à ampla abrangência desse tema, este projeto se restringirá a

	entre outros, e sua definição deverá levar em conta essas modalidades de áudio.	definir com detalhes apenas a representação de sons musicais.
	Em cada fase desse processo são gerados modelos (diagramáticos ou textuais) que visam a representar os requisitos do software naquela fase.	**Porém**, com essa diversidade de métodos não existe um padrão de modelos para as atividades do desenvolvimento de software.
	Apesar da idéia de processar informações paralelamente ser antiga,	**só** nos últimos anos começou realmente a se desenvolver, sendo que atualmente esta área é alvo de intenso estudo dentro da comunidade acadêmica.
	Ao invés de memorizar informação,	os estudantes devem ser ensinados a buscar e a usar a informação (...)
	Apesar dessa prática profissional ter sido desenvolvida principalmente orientada a procedimentos,	foi usada orientação a objetos em outros trabalhos, como um relatório técnico do ICMC e dois artigos em congressos internacionais dos quais sou co-autora.
	Embora eles possam ser tão simples como subrotinas,	geralmente eles são modelados como entidades maiores contendo algum tipo de controle persistente e, principalmente, autonomia.
	Independentemente da qualidade da concepção, desenvolvimento e teste do sistema antes de ter sido liberado,	o produto de software irá certamente ser modificado por diversas razões (...)
	Mesmo existindo diversos paradigmas de desenvolvimento,	essas atividades são organizadas em três fases genéricas, independentemente da área de aplicação, tamanho ou complexidade do projeto.
	Mesmo quando essas empresas tentam iniciar a melhoria de processo usando um desses modelos,	elas esbarram em um empecilho muito grande.
	Muito embora o termo qualidade seja vago e subjetivo,	este trabalho adota qualidade como sinônimo de adequação para uso (fitness for use), entendendo por isso, a característica do produto que contempla as expectativas e necessidades do usuário.
	Por maiores que sejam os problemas, e por mais sofisticado que seja o sistema ou a instalação onde ele é processado,	a atividade de manutenção não pode ser evitada.
	Qualquer que seja o tipo de manutenção - corretiva, evolutiva, adaptativa ou preventiva -	algumas tarefas comuns devem ser efetuadas: o entendimento, a modificação e a revalidação do software.

Tabela D.8 – Exemplos e marcadores superficiais para a relação CONCLUSION

Ordem	Proposição 1	Proposição 2
NS	Por serem modelos matemáticos, com processo de aprendizado através da atualização de pesos com a apresentação sucessiva de exemplos de treinamento, as redes neurais são capazes de generalizar para reconhecer novos exemplos, adaptando-os a novas classes com sucesso. (...)	**Assim**, os modelos conexionistas provêm mecanismos mais gerais para inferência, desde que se manipule corretamente as redes para cada exemplo apresentado.
	Qualquer que seja a abordagem utilizada	**Dessa forma**, é desejável que durante o

	para o desenvolvimento de uma aplicação hipermídia, é necessário elaborar uma estrutura para as informações. (...)	desenvolvimento de uma aplicação hipermídia para ensino, o autor consiga depreender com clareza a estrutura da mesma, para que ele possa organizar as informações conforme as estratégias pedagógicas adotadas, de maneira a atingir os seus objetivos (e também os do aprendiz). (...)
	Segundo Thüring, existem dois tipos de aplicações hipermídia no que se refere ao relacionamento entre estas e o conhecimento. O primeiro tipo encoraja aqueles que querem navegar por grandes volumes de informação, obtendo informação ao longo do caminho. Este tipo é mais apropriado para suportar buscas irrestritas e recuperação de informação. O segundo tipo está diretamente relacionado a problemas e soluções específicas, e é totalmente estruturado e restrito. Este tipo é mais adequado para tarefas que requerem um profundo entendimento e aprendizagem.	**Desse modo**, uma aplicação hipermídia bem projetada deve facilitar a sua própria compreensão, sendo totalmente coerente e evitando a desorientação do leitor.
	Essa camada central é também conhecida por máquina de hipertexto (hypertext engine) e corresponde tanto ao nível da máquina HAM quanto à camada de armazenamento em Dexter.	**Este é o ponto diferencial** para que outras aplicações façam uso da tecnologia de hipertextos sem maiores esforços na implementação.
	O conjunto de operações que podem ser definidas sobre imagens é completamente disjunto do conjunto de operações de busca e comparação de imagens (que pode e deve ser utilizado para implementar e otimizar as operações de busca do banco de dados).	Tais operações são, **portanto**, distintas das que tradicionalmente são suportadas para os tipos mais convencionais de dados.
	Os Sistemas Computacionais Distribuídos, há mais de uma década, já deixaram de ser apenas uma promessa, não estando mais restritos a pesquisas realizadas nos meios acadêmicos. (...)	**Sendo assim, é importante** que existam boas ferramentas de apoio à construção dessas aplicações, procurando-se tornar seu trabalho de implementação semelhante ao das aplicações centralizadas, cuja técnica já é amplamente conhecida e dominada pela maioria dos usuários (programadores).

Tabela D.9 – Exemplos e marcadores superficiais para a relação CONDITION

Ordem	Proposição 1	Proposição 2
NS	Um método proposto por Bernstein para descobrir automaticamente os links a partir do conteúdo entre unidades relacionadas em um hipertexto de monografias, enfatiza que um hipertexto é inútil	**caso** não forneça aos leitores uma coleção significativa de links.
	Assim, os modelos conexionistas provêm mecanismos mais gerais para inferência,	**desde que** se manipule corretamente as redes para cada exemplo apresentado.
	Um resultado mais efetivo pode ser obtido	**se** for possível estabelecer um equilíbrio

		entre o controle do aprendiz e do sistema, de modo a oferecer um certo grau de orientação para que o aprendiz possa atingir seu objetivo sem perder a flexibilidade da leitura.
SN	**Caso** não seja verificada uma ocorrência,	o fato é notificado ao usuário para que ele possa optar pela ação desejada, como por exemplo, ignorar o fato, corrigir a palavra ou acrescenta-la ao seu dicionário pessoal.
	dependendo da granularidade escolhida,	essa unidade pode ser tanto um módulo quanto um procedimento do programa (...)
	Se aplicados nas fases iniciais do desenvolvimento do software,	eles podem detectar erros que só seriam descobertos nas fases de teste e depuração (...)

Tabela D.10 – Exemplos e marcadores superficiais para a relação ELABORATION

Ordem	Proposição 1	Proposição 2
NS	No caso de aplicações hipermídia para ensino, é desejável uma estrutura que inclua o material essencial ou relevante, sem eliminar o secundário, o complementar, desde que este não seja conflitante com o anterior e que possa contribuir para os objetivos do aprendiz.	A estrutura **também** deve incluir informações que representem as estratégias pedagógicas utilizadas pela aplicação.
	O Grupo de Engenharia de Software do Instituto de Ciências Matemáticas e de Computação- ICMC/USP, em colaboração com o Grupo de Engenharia de Software da Faculdade de Engenharia Elétrica da UNICAMP, tem desenvolvido pesquisas na área de teste, com ênfase em estudos teóricos e empíricos e no desenvolvimento de ferramentas de teste. (...)	**A exemplo do que ocorre** na atividade de teste, durante o desenvolvimento de software, várias restrições são impostas à atividade de teste de regressão: custo, tempo, pressões de mercado, e outras. (...)
	Recentemente, a aplicação da informática em ambientes de ensino e aprendizagem tem sido alvo de intensas pesquisas.	**A grande maioria desses trabalhos** aponta para a necessidade de se romper as fronteiras da sala de aula convencional e oferecer aos professores a oportunidade de trabalhar seus conteúdos programáticos, proporcionando aos estudantes uma confortável e eficiente construção do conhecimento.
	A criação de algoritmos que implementem Redes Neurais Construtivas, chamados Algoritmos Construtivos, é recente nas pesquisas em Redes Neurais Artificiais.	**A implementação destes algoritmos** normalmente é restrita às preferências do projetista, sendo muitas vezes dependente de plataformas e formato de dados.
	Esse critério apóia-se em duas hipóteses: Programador Competente e Efeito de Acoplamento.	**A primeira** afirma que um programa produzido por um programador competente, ou está correto ou próximo do correto.
	A estratégia de minimização proposta foi desenvolvida para a ferramenta de teste Proteum,	**a qual apóia** o critério Análise de Mutantes.
	Neste contexto, este projeto tem como objetivo definir uma nova classe de tipos de	**a qual pode ter** as operações de busca e comparação adaptadas aos requisitos

	dados,	dessa classe.
	O revisor DTS também é uma ferramenta mais completa de revisão,	**abordando** quatro classes principais de erros: a revisão ortográfica; a revisão de estrutura, que localiza erros como falta de pontuação, uso desbalanceado de delimitadores, uso de palavras repetidas, etc.; a revisão gramatical que trata, entre outros, de erros de concordância nominal e verbal e uso indevido de crase; e a revisão de estilo, que verifica uso correto de pronome e da partícula 'se' e também de períodos muito longos.
	Esse processo define as atividades de desenvolvimento de software,	**abrangendo** as pessoas e as suas responsabilidades, os recursos, os cronogramas, os orçamentos, o processo de desenvolvimento em si, as técnicas, os métodos, as ferramentas e os outros elementos relacionados ao software.
	Segundo Lange, para que as aplicações hipermídia sejam “executadas” em diferentes sistemas elas devem ser especificadas através de modelos formais e abstratos.	**Adicionalmente**, modelos formais adequados podem oferecer abordagens sistemáticas e confiáveis para analisar e verificar propriedades estruturais e dinâmicas de aplicações hipermídia.
	do ponto de vista de qualidade do processo, o teste sistemático é uma atividade fundamental para a ascensão ao Nível 3 do Modelo CMM 2 do SEI.	**Ainda**, o conjunto de informação oriundo da atividade de teste é significativo para as atividades de depuração, manutenção e estimativa de confiabilidade de software.
	Este trabalho apresenta o Método para Projeto de Hiperdocumentos para Ensino, ou EHDM (Educational Hyperdocuments Design Method), que foi definido inicialmente como uma ferramenta de modelagem para o SASHE, mas que posteriormente mostrou-se útil para apoiar o desenvolvimento de hiperdocumentos educacionais para outros sistemas e/ou ambientes, por exemplo, para a World Wide Web.	**Além da** definição do EHDM, este trabalho também apresenta o protótipo da ferramenta que suporta o método definido e um exemplo de sua utilização, permitindo avaliar a possibilidade de uso do EHDM num contexto real. (...)
	A utilização de ferramentas que procuram automatizar essa atividade propicia uma maior eficácia e uma redução do esforço necessário para a sua realização,	**além de** diminuir os erros nesta atividade decorrentes da intervenção humana.
	Um problema relacionado à computação paralela é o alto custo de aquisição e manutenção de arquiteturas paralelas.	**Além disso**, a compra de uma arquitetura geralmente implica na dependência do comprador ao fabricante.
	Existem vários tipos de sensores que podem ser utilizados para medição de distâncias.	**Alguns deles** são simples, servindo somente para detectar objetos mais ou menos próximos, medir distâncias e, possivelmente, ângulos.
	e vêm adotando uma série de aperfeiçoamentos, que visam a obtenção de um melhor desempenho.	**Alguns exemplos** de abordagens utilizadas para enriquecer as arquiteturas de von Neumann são as diferentes formas de pipeline, as unidades funcionais, processador vetorial, etc.

	A grande revolução que está ocorrendo na área de informática, proveniente de conseqüentes aprimoramentos tanto em hardware como em software, tem contribuído para a sedimentação dos sistemas hipertexto/hipermídia.	**Aliado a este fator**, há a massificação da utilização da Internet. (...)
	que considera o hardware (a taxonomia de Duncan foi considerada) com seus elementos básicos (processadores, memória, canais de comunicação, etc.) e o software.	**Ambos** são modelados a partir da teoria dos grafos e segundo o paradigma da orientação a objetos. (...)
	Nos Sistemas de Exploração Livre o estudante navega em um documento, onde as informações estão organizadas sob a forma de páginas, conectadas através de links (elos de ligação).	**Aqui**, o controle é inteiramente do usuário que decide qual caminho deseja seguir, quando voltar ou pular adiante, quando seguir um determinado caminho e quando evitar as distrações com possíveis caminhos irrelevantes. (...)
	A principal característica deste framework é a distinção, em um documento hipermídia, dos aspectos de conteúdo e estrutura por um lado, e dos aspectos de armazenamento e tempo de execução por outro.	**Aspectos estes** que devem ser levados em consideração na construção de um modelo formal para especificação de aplicações em sistemas hipermídia abertos, que é o alvo desse trabalho de mestrado.
	onde todo o programa é testado novamente,	**até mesmo** as partes do programa que não foram afetadas pelas modificações.
	onde grandes conjuntos de teste são gerados	**baseando-se** apenas no domínio de entrada do programa.
	Os autores devem estar registrados em algum grupo para terem acesso à edição das páginas.	**Cada grupo** tem um conjunto de páginas que lhe é específico.
	Os tutoriais geralmente possuem uma estrutura seqüencial rígida,	**com** o programa apresentando informações sobre um assunto e, em seguida, fazendo uma série de perguntas a respeito. (...)
	e que apoiam os critérios Baseados em Fluxo de Dados,	**como é o caso** da Poke-Tool, Atac e Asset
	Um Sistema Reativo é um programa de computador que mantém um interação permanente com seu ambiente externo, o qual pode ser um usuário, um dispositivo de entrada ou uma outra parte do sistema. (...)	**Como exemplos** desses sistemas, podem-se citar controle de tráfego aéreo, controle metroviário e controle de monitoramento hospitalar.
	As metodologias suportam a criação dessas aplicações de modo mais genérico	**como, por exemplo**, apoiando as fases de modelagem de navegação e de interface.
	e este realiza operações	**como** salvar e recuperar versões de uma página.
	Logo, a flexibilidade oferecida pode levar o usuário a se perder no hiperdocumento	**como também** dificultar o encontro das informações desejadas.
	Surgiram, portanto, métodos que pudessem auxiliar essa tarefa,	**considerando-se** diferentes perspectivas: dos usuários, dos desenvolvedores e da organização.
	Dentre as técnicas de verificação e validação, a atividade de teste é uma das mais utilizadas,	**constituindo** um dos elementos para fornecer evidências da confiabilidade do software em complemento a outras atividades,
	nas quais usuários navegam	**contendo** ligações embutidas em seu

	interativamente por documentos	conteúdo
	O desenvolvimento de sistemas de informação, em qualquer tipo de organização, tem utilizado cada vez mais o suporte de Sistemas de Gerenciamento de Bases de Dados,	**cuja** frente tecnológica atual apóia-se nos Gerenciadores Relacionais, no projeto e no desenvolvimento baseados no paradigma de Orientação a Objetos.
	Dentre os ambientes de programação via troca de mensagens, destacam-se as plataformas de portabilidade, (destinados a possibilitar o transporte de programas paralelos entre plataformas computacionais distintas)	**das quais** dois representantes merecem destaque no cenário computacional atual: o MPI e o PVM.
	Com a Verificação busca-se garantir que o produto está sendo desenvolvido da forma correta, ao passo que com a Validação tenta-se se assegurar que o produto que está sendo desenvolvido está em conformidade com as expectativas do cliente.	**Dentre** as atividades de Verificação e Validação, o teste é uma das atividades mais utilizadas, sendo de grande importância para a identificação e eliminação de erros no produto. (...)
	A flexibilidade no acesso à informação, junto com boas capacidades de navegação em hipertextos, fizeram com que estes sistemas fossem bastante atrativos para a utilização como programas de recuperação de informação.	**Dentro desta visão**, quanto mais controle o usuário puder ter, maior chance de sucesso haverá na interação com o sistema.
	Assim, o GEO define diversas características, cuja principal motivação é classificar tipos de dados não usuais quanto a operações que podem ser utilizadas para essa manipulação.	**Dessa forma, por exemplo,** a característica de imagens refere-se a um tipo de dado que permite a armazenagem e manipulação de um atributo imagem.
	Um modelo recente proposto por Schwabe et al consiste de sistemática de projeto baseada no modelo OOHDM (Object-Oriented Hypermedia Design Model). (...)	**Deve-se observar que** o OOHDM já considera a implementação de hiperdocumentos em HTML, ou seja, de um servidor de WWW (World-Wide Web).
	Diversas técnicas vêm sendo propostas	**e** baseiam-se em duas abordagens: as de técnicas de aferição, onde as informações requeridas por um estudo podem ser obtidas a partir do próprio sistema, e as técnicas de modelamento onde se constrói um modelo representativo do sistema. As técnicas de aferição podem ser: coleta de dados, benchmarks ou construção de protótipos. (...)
	Assim, este trabalho tem como ponto de partida a avaliação de como estruturas de modelagem podem evoluir, preservando propriedades que possam contribuir para uma integração dinâmica dos esquemas de duas bases de dados construídas isoladamente, mas a partir de uma mesma BTO.	**É bom salientar** que essas ferramentas se apóiam nas técnicas de projeto associadas ao paradigma de orientação a objetos, o que corresponde hoje à tendência predominante para o desenvolvimento de novos aplicativos.
	O sistema funciona como corretor ortográfico, detectando erros, dando sugestões de correção e permitindo inserir	**É possível também** fazer pesquisa aproximada de uma dada palavra no dicionário.

	novas palavras em um dicionário pessoal.	
	Segundo Peterson, a técnica Redes de Petri foi desenvolvida para modelar sistemas com interação de componentes paralelos e concorrentes.	**Ela** é um modelo formal e abstrato do fluxo de informação, representado através de um grafo que modela as propriedades estáticas do sistema e que, através de sua execução, permite também representar as propriedades dinâmicas.
	Além disso, como parte de revisão bibliográfica, foram estudados também alguns modelos de versão de software para SCM (Software Configuration Management) juntamente com alguns conceitos básicos relacionados ao controle de versão tais como: definição de uma versão, como são geradas as diversas versões (ou revisões) de um sistema (ou de um software), como são armazenadas todas essas versões em um espaço mínimo de armazenamento, geração de branches, além de vários outros conceitos relacionados ao controle de versão.	**Em adição**, para viabilizar o desenvolvimento da ferramenta, foram estudados alguns mecanismos para programação na Web tais como CGI (Common Gateway Interface), Java, Applets, Servlets e JSP.
	Por exemplo, os desenvolvedores de software freqüentemente fazem mudanças em módulos de software quando erros são detectados, e os clientes raramente ficam satisfeitos com a primeira versão fazendo, então, com que várias revisões sejam produzidas antes de produzir uma versão final.	**Em cada um desses casos**, objetos em desenvolvimento são alterados e atualizados de forma a produzir o próximo refinamento no processo evolucionário.
	Podem-se agrupar os critérios de teste em três técnicas: Técnica Funcional, Técnica Estrutural e Técnica Baseada em Erros.	**Em nível de** programa, os critérios de teste existentes são complementares e devem ser aplicados em conjunto, aumentando dessa maneira a qualidade da atividade de teste.
	As duas principais operações que devem ser suportadas eficiente e adequadamente por um SGBD são a consulta e a manipulação de dados.	**Em particular**, em sistemas que suportam meta-estruturas, tal como é o caso do GEO, essas operações podem ser feitas tanto no esquema de dados quanto no banco de dados propriamente dita.
	Assim pode-se ter desde padrões de análise,	**em que** esse nível é bastante alto
	No entanto, existem diversos tipos de aplicações multimídia distribuídas que não usam os serviços da WWW.	**Entre eles** pode-se citar aplicações de vídeo sob demanda, jogos em rede e a TV interativa ou TV digital.
	Este trabalho dá continuidade à atividade de desenvolvimento de ferramentas de teste e validação para Sistemas Reativos, com ênfase no critério Análise de Mutantes.	**Especificamente**, desenvolve a ProteumRS/ST, versão Interface e versão Script, para apoiar a validação de especificações baseadas em Statecharts, a partir das definições e especificações conduzidas por Fabbri.
	Visto isso, é proposta neste trabalho uma abordagem alternativa.	**Essa abordagem** não se baseia em um léxico para determinação das classes sintáticas das palavras. (...)
	Os métodos formais permitem que se	**Exemplos de** técnicas utilizadas pelos

	especifique, desenvolva e verifique o software ou parte dele, de modo sistemático.	métodos formais são: Máquinas de Estados Finitos (MEFs), Redes de Petri e Statecharts.
	Sistemas desta última classe, os Sistemas Reativos, caracterizam-se por interagir continuamente com o ambiente, reagindo a eventos externos gerados pelo processo controlado.	**Incluem-se nessa classe**, Sistemas de Tempo Real, Sistemas Embutidos e Sistemas Críticos com relação à segurança. (...)
	Nessa definição, é revisto cada um dos subsistemas componentes,	**inclusive** aqueles que podem ser obtidos comercialmente.
	Por outro lado, a técnica estrutural, ou teste da caixa branca, é baseada no conhecimento da estrutura interna da implementação,	**mais especificamente** no fluxo de controle e em informações do fluxo de dados necessárias para derivar os requisitos de teste.
	Com a diminuição do tamanho físico dos equipamentos de rádio, da instrumentação e dos equipamentos de fotografia convencional ou digital, é possível a substituição do avião convencional pelo aeromodelo com grande economia no custo inicial do sistema.	**Mais que isso**, os custos de manutenção e do piloto são reduzidos consideravelmente, pois no caso de um aeromodelo qualquer pessoa pode ser treinada para operá-lo em um curto espaço de tempo.
	Diversas áreas da ciência e engenharia precisam de sistemas que capturem, processem e integrem informações provenientes de várias fontes.	**Muitos destes sistemas** interagem com o mundo real e requerem informações precisas e confiáveis do ambiente ao seu redor. (...)
	Observa-se a relação com a hipótese do Programador Competente em nível de programa,	**na qual** apóia-se o critério Análise de Mutantes.
	Diversas possibilidades de implementação do sistema são investigadas e definidas neste trabalho, com níveis crescentes de complexidade e funcionalidade.	**Nessa definição**, é revisto cada um dos subsistemas componentes, inclusive aqueles que podem ser obtidos comercialmente. (...)
	É bom salientar que essas ferramentas se apóiam nas técnicas de projeto associadas ao paradigma de orientação a objetos,	**o que corresponde** hoje à tendência predominante para o desenvolvimento de novos aplicativos.
	Outro fato que causa redundância é a geração aleatória,	**onde** grandes conjuntos de teste são gerados baseando-se apenas no domínio de entrada do programa.
	Estratégias que realizam a minimização de conjuntos de casos de teste permitem reduzir os custos asssociados a atividade de teste, mais particularmente, ao critério em questão. Existem algumas estratégias propostas que visam a obtenção do menor conjunto possível a partir de um conjunto inicial, buscando, desta forma, a máxima redução.	**Outra vantagem** da minimização é a possibilidade de fornecer parâmetros mais reais para quantificar o custo de aplicação de um critério durante o desenvolvimento de estudos empíricos.
	Entretanto, ainda não existe nenhuma estratégia desenvolvida para minimização de conjuntos adequados ao critério Análise de Mutantes, principalmente para o teste de programas na linguagem C.	**Para essa linguagem**, a única ferramenta de apoio ao critério Análise de Mutantes é a ferramenta Proteum, a qual não trata o problema da minimização.
	Dispondo desse recurso, é possível definir-se objetos no GEO que tenham como	**Por exemplo**, pode-se definir um objeto do tipo pássaro com os atributos do tipo string

	atributos, além daqueles comumente utilizados tais como inteiros e strings, também atributos cujo valor é um som.	nome e país-de-origem, e os atributos do tipo áudio, como canto-de-chamada e sinal-de-perigo.
	Problemas com a escrita podem afetar o desempenho de profissionais de maneira marcante,	**principalmente** no caso de pesquisadores e acadêmicos que precisam escrever com proficiência e desembaraço não apenas na língua materna, mas também em uma ou mais línguas estrangeiras.
	Gamma apresenta padrões de projeto de utilização bastante ampla	**que** abrange sistemas os mais diversos
	No primeiro foi usado o método JSD,	**que** é um método estruturado com base nos dados
	Muitas das técnicas de teste de regressão utilizam mecanismos, técnicas e critérios oriundos da atividade de teste de software realizada durante o processo de desenvolvimento.	**São exemplos** dessas técnicas: Técnica baseada em Fluxo de Dados e Técnica baseada em Mutação Seletiva.
	Este modelo deverá atender, principalmente, aos requisitos propostos por Osterbye e Wiil	**seguindo** o framework denominado FLAG para sistemas hipermídia abertos.
	são necessárias atividades de garantia de qualidade, tais como verificação e validação,	**sendo** a atividade de teste uma das mais utilizadas, constituindo-se em um dos elementos para fornecer evidências da confiabilidade do software em complemento a outras atividades.
	e trata do ensino da escrita técnica para uma comunidade de pesquisa específica, a CHI (Conference on Human Factors in Computing Systems),	**sendo que** a seção escolhida para a análise nesse trabalho foi a Introdução.
	Um Sistema Reativo é um programa de computador que mantém um interação permanente com seu ambiente externo, o qual pode ser um usuário, um dispositivo de entrada ou uma outra parte do sistema.	**Seu comportamento** é baseado na relação entre eventos de entrada e de saída que ocorrem discretamente no tempo.
	é necessário fornecer, como entrada para elas, um corpus já etiquetado com marcas chamadas, no inglês, de part-of-speech tags.	**Tais marcas** ou etiquetas são, principalmente, as categorias gramaticais (morfossintáticas) das palavras do corpus.
	Segundo Garzotto, o maior desafio encontrado pelos autores de aplicações hipermídia é capturar e organizar assuntos complexos de maneira adequada, facilitando posteriormente a sua manutenção. (...)	Garzotto **também** afirma que uma abordagem estruturada para o desenvolvimento de tais aplicações sugere a noção de authoring-in-the-large, que permite a descrição de classes gerais de informação e estruturas navegacionais e de authoring-in-the-small que se refere ao desenvolvimento do conteúdo dos nós. (...)
	Large, por exemplo, diz que realizar ligações entre as informações do documento pode ser necessário para o aprendizado, porém, não é o suficiente, pois não se levam em conta fatores como a idade, a habilidade, a experiência anterior do usuário tanto quanto ao domínio, como quanto ao uso de tais sistemas.	**Também se argumenta** que existe pouca evidência empírica que mostre uma contribuição educacional relevante na utilização da liberdade total.

	Trabalhos correlatos para a língua inglesa, por exemplo, já deram origem a várias ferramentas computacionais de realização lingüística.	**Todas elas** tratam da geração de sentenças, ou seja, deixam de fora problemas complexos de geração de texto, como o tratamento de discurso.
	Em paralelo, existem diversos trabalhos que buscam entender e classificar os defeitos que ocorrem em sistemas de software.	**Um desses trabalhos** é a taxonomia de DeMillo/Mathur que classifica os defeitos que ocorrem durante a fase de codificação do software.
	Desde sua introdução, sistemas hipermídia têm sido utilizados em ambientes de apoio ao ensino.	**Um exemplo** é o Sistema Intermedia, desenvolvido na Brown University nos anos 80, utilizado como ferramenta de apoio em cursos de Literatura e Biologia, entre outros.
	Também foram criadas novas estruturas inseridas no ambiente Hip/Windows visando o auxílio à autoria, no que se refere à criação de roteiros, criação de nós com funções específicas, classificação de nós terminais, etc.	**Um outro conceito** utilizado para a autoria é o de reutilização de objetos. (...)
	Dispondo desse recurso, é possível definir-se objetos no GEO que tenham como atributos, além daqueles comumente utilizados tais como inteiros e strings, também atributos cujo valor é um som. (...)	**Vale ressaltar que** este trabalho visa dar suporte dentro de um banco de dados, a informações não tradicionais, permitindo a armazenagem e recuperação de dados com a característica de áudio num banco de dados GEO. (...)
SN	**Além de** aumentar o número de possíveis usuários,	a elaboração de uma máquina paralela virtual sobre o Windows95 permitirá também a união de vários projetos de pesquisa do grupo de Programação Concorrente e Sistemas Distribuídos do ICMSC/USP, referentes à simulação distribuída, balanceamento de cargas e ferramentas para o desenvolvimento de programas paralelos, todos desenvolvidos para uma plataforma com Computadores Pessoais e com o Windows.
	Definida como uma aplicação de SGML,	HTML (Hypertext MarkUp Language) é uma linguagem que fez sucesso por sua simplicidade.

Tabela D.11 – Exemplos e marcadores superficiais para a relação ENABLEMENT

Ordem	Proposição 1	Proposição 2
NS	Por outro lado, os jogos instrucionais podem ser utilizados para a criação de situações que favorecem a aprendizagem.	**Com isso**, é relevante a participação do professor para influenciar o estudante a utilizar esse recurso para aprender sobre algum assunto específico.
	Logo, este trabalho foi proposto de forma a colaborar na execução dessas atividades,	**fornecendo** um conjunto de requisitos que possua as características mencionadas anteriormente.
	A maioria desses trabalhos aponta para a necessidade de se romper as fronteiras da sala de aula convencional,	**oferecendo** aos professores a oportunidade de tornar seus conteúdos mais dinâmicos e proporcionando aos estudantes uma nova forma de construção do conhecimento.

	e para que ela continue crescendo é necessário que exista um gerenciamento apropriado das suas informações.	**Para alcançar esse objetivo** têm sido utilizados novos métodos de acompanhamento e monitoramento dos problemas agrícolas, tais como, a fotografia aérea e o sensoriamento remoto por satélite. (...)
	O fato de se ter um programa testado não garante a ausência de defeitos, o que não permite afirmar que o programa está correto.	**Para apoiar** a fase de teste de software, vários métodos, técnicas e ferramentas têm sido propostos, os quais contribuem para a sistematização e para o aprimoramento da qualidade dessa atividade e, conseqüentemente, da qualidade final do produto em desenvolvimento.
	Os sistemas distribuídos aplicados à computação paralela foram criados, portanto, para permitir uma melhor relação custo/benefício para a computação paralela, pois oferecem a potência computacional necessária a uma grande quantidade de aplicações.	**Para viabilizar** essa idéia foram desenvolvidas ferramentas de software que permitem a utilização do conceito de máquina paralela virtual. (...)
SN	Assim, parte-se da suposição que, nestes dados, possa existir alguma forma de definição primitiva para os diversos elementos que devem ser compartilhados,	**a partir do qual** sua instanciação em elementos de um esquema de dados numa organização em particular possa ser reconhecido.
	Inicialmente, será definido um esquema no qual existirá o tipo som, juntamente com seus atributos e relacionamentos. Na definição de atributos serão ainda definidas as possíveis restrições que esses possam ter.	**Através desta definição**, será possível o armazenamento de som no MRO, permitindo a recuperação do som desejado. (...)
	Procurando estender a aplicação do critério Análise de Mutantes para o teste de integração, um novo critério denominado Mutação de Interface (Interface Mutation-IM) foi desenvolvido.	**Com esse critério é possível** testar a interface entre as unidades que compõem o software, ao contrário da Análise de Mutantes, que explora as características das unidades separadamente.
	Além disso, um estudo preliminar realizado por Wong et al, comparando a Mutação Restrita no contexto das linguagens C e Fortran, resultou na seleção de um subconjunto de operadores de mutação da ferramenta Proteum,	**constituindo** uma base para a determinação do conjunto essencial de operadores da linguagem C.
	um ponto crucial que se coloca nessa perspectiva é a escolha e/ou a determinação de uma estratégia de teste, que em última análise passa pela escolha de critérios de teste,	**de forma que** as vantagens de cada um desses critérios sejam combinadas objetivando uma atividade de teste de maior qualidade.
	As aplicações tradicionais de banco de dados incluem aqueles sistemas cuja estrutura dos dados tratados são intrinsicamente homogêneas. (...)	**Dispondo desse recurso, é possível** definir-se objetos no GEO que tenham como atributos, além daqueles comumente utilizados tais como inteiros e strings, também atributos cujo valor é um som. (...)
	Este trabalho propõe justamente estender o SIRIUS neste aspecto,	**e com isso** incrementar sua utilidade prática.
	Os critérios de teste estabelecem requisitos	**e, através** da análise da satisfação desses

	que devem ser cumpridos	requisitos, **consegue-se** uma maneira de quantificar a atividade de teste.
	Ambos são modelados a partir da teoria dos grafos e segundo o paradigma da orientação a objetos. Assim, software e hardware são reduzidos a um grafo e o modelo global produzido é totalmente orientado a objetos.	**Isso permite** o estabelecimento da ferramenta F.A.P.P., que utiliza a metodologia de desenvolvimento de programas desenvolvida por Foster, que propõe um desenvolvimento análogo ao utilizado em programas seqüenciais, seguindo a técnica de programação por esqueleto proposta por Zima e Chapman.
	Uma característica do modelo é que ele presta-se bem à implementação de um Gerenciador de Objetos que o suporte, sendo que seu formalismo apresenta indicações de como essa implementação deve ser realizada,	**o que oferece** uma indicação também formal de como uma modelagem feita em SIRIUS pode ser mapeada para implementação em um outro modelo.
	Os sistemas hipermídia respondem estas perguntas na hora em que elas são geradas,	**oferecendo** a possibilidade e a facilidade da exploração do documento a ser estudado.
	Cada modelo de RNC possui seu próprio critério de inserção de novos neurônios e conexões	**A diferença entre estes critérios permite** a construção de redes diferentes para um mesmo problema.
	Através desta definição, será possível o armazenamento de som no MRO,	**permitindo** a recuperação do som desejado.
	O sistema apenas fornece o material e proporciona uma forma de navegação através dele, com o controle da interação totalmente a cargo do aprendiz,	**permitindo que** este tenha progresso de acordo com os seus interesses e objetivos.
	Esta análise é feita através da utilização de benchmarks e de um exemplo de aplicação paralela,	**possibilitando** estudar o comportamento do MPI em determinadas situações.
	Os testes foram realizados em três implementações do MPI de domínio público que executam sobre a plataforma LINUX.	**Tal procedimento possibilita** uma análise comparativa entre as três implementações, a fim de se determinar, por exemplo, até que ponto uma especificação centrada na eficiência pode garanti-la em qualquer implementação.
	Desta forma, está mais fácil manipular as informações armazenadas em diferentes mídias,	**viabilizando** cada vez mais sua aplicação nas mais variadas áreas.

Tabela D.12 – Exemplos e marcadores superficiais para a relação EXPLANATION

Ordem	Proposição 1	Proposição 2
NS	e é calculado o índice de legibilidade,	**isto é**, um indicador de dificuldade de entendimento do texto.
	No entanto, este trabalho não se insere como pesquisa sobre o tratamento de informações multimídia,	**pois** não leva em consideração o suporte à temporização e sincronização entre múltiplos atributos com características de áudio (ou mesmo imagens e gráficos).
	o que de uma certa forma é considerado vantagem	**por** não permitir que o estudante se desvie do objetivo inicial.
	A utilização de um modelo de representação para o domínio da aplicação	**porque** permite que ele trabalhe em um nível de abstração mais próximo a este

	pode auxiliar o autor na atividade de autoria	domínio, entre outras vantagens.
	Foi selecionado como domínio para um protótipo de aplicação do SIATE o tópico Aquisição de Conhecimento para Sistemas Inteligentes, onde são abrangidos os subtópicos de Aquisição de Conhecimento Explícito e Implícito.	**Esta seleção se deve** ao gargalo existente no processo de aquisição de conhecimento especialista.
	As técnicas de teste devem ser vistas como complementares,	**sendo que** a questão está em como empregá-las de forma que as vantagens de cada uma delas resultem na determinação de uma atividade de teste de melhor qualidade.

Tabela D.13 – Exemplos e marcadores superficiais para a relação EVALUATION

Ordem	Proposição 1	Proposição 2
NS	Um sistema baseado em aeromodelos e equipamentos de rádio controle convencionais foi desenvolvido para a avaliação do emprego da tecnologia no monitoramento de problemas agrícolas.	**Os resultados obtidos são altamente adequados** à utilização do sistema em várias aplicações, embora algumas delas necessitem de características especiais que podem ser atendidas em futuras implementações do sistema.
	A realização de estudos empíricos intensificou-se nos últimos anos procurando avaliar as diferentes técnicas e critérios de teste existentes, de modo a definir uma estratégia confiável e de baixo custo para a realização da atividade de teste, em que o custo, a eficácia e a dificuldade de satisfação (strength) são fatores básicos para comparar a adequação de um critério de teste. (...)	Com a proposição do critério Mutação de Interface é evidente o **aspecto positivo** de se utilizar o mesmo conceito de mutação nas diversas fases do teste; é também evidente a indagação sobre qual estratégia utilizar para obter-se a **melhor relação custo/eficácia** quando são aplicados os critérios Análise de Mutantes e Mutação de Interface no teste de um produto. A proposta de trabalho apresentada neste texto coloca-se nesta perspectiva.
	Dessa forma, considerando-se a concentração de trabalhos relacionados ao desenvolvimento de sistemas hipermídia e as limitações observadas nos trabalhos revisados relacionados à definição de requisitos para o desenvolvimento desses sistemas, este trabalho propõe um conjunto único de requisitos, que possui as propriedades de ser abrangente e suficientemente completo para auxiliar a etapa de engenharia de requisitos de um novo sistema ou para permitir a avaliação de um sistema já existente. (...)	**Os benefícios advindos** a partir da realização deste trabalho estão relacionados principalmente à proposta de um conjunto de requisitos para sistemas de autoria hipermídia educacional e à obtenção de informações referentes à qualidade da implementação do SASHE.
	As soluções mais exploradas são as que procuram diminuir o número de mutantes a serem executados e analisados;	**os resultados obtidos demonstram** que é possível reduzir sensivelmente o número de mutantes gerados sem comprometer a **eficácia** do critério em revelar a presença de erros.
	Duas dessas abordagens são a Mutação Aleatória e a Mutação Restrita. (...)	**Estudos demonstram** que a utilização dessas abordagens reduz significativamente o número de mutantes

		gerados, **sem ocasionar grandes perdas na eficácia** em revelar a presença de erros do critério.
	Embora o termo desempenho esteja diretamente relacionado com a computação paralela, o objetivo inicial deste trabalho é permitir a utilização da computação paralela no ambiente Windows95, mesmo que para isso o desempenho fique um pouco abaixo do PVM utilizado no UNIX.	Uma situação de **empate**, nesta etapa, já seria um **ótimo resultado**, porém, não foi descartada a busca por um **melhor desempenho** durante o desenvolvimento do trabalho, apenas essa busca não foi considerada como objetivo fundamental.
	Devido a crescente importância que as redes de computadores tem adquirido, é essencial uma ferramenta de simulação de redes de computadores para o ASiA.	A **importância** do sistema para simulação de redes do ASiA está relacionada com a grande **flexibilidade** apresentada pelo ASiA e pela possibilidade de se integrar a simulação de redes com outras **facilidades** disponíveis nesse sistema, tais como: simulação de arquiteturas de computadores; utilização de ferramentas para análise da saída; utilização de ferramentas para visualização dos resultados, etc.
	o simulador para Redes Neurais Artificiais chamado Kipu, desenvolvido em um trabalho de Mestrado, foi avaliado,	constatando-se a **necessidade** do desenvolvimento de uma versão **mais eficiente e flexível**.
	Os experimentos apresentados neste trabalho utilizaram os Algoritmos Construtivos	que foram revisados e **utilizados com sucesso** no treinamento de Redes Neurais com os conjuntos de padrões selecionados.
	A tecnologia de agentes vem também facilitar a criação de software capaz de interoperar em ambientes heterogêneos.	Além das **vantagens** apresentadas acima, temos uma **maior flexibilidade** e **adequação** para uso em um modelo de implementação cliente/servidor. (...)

Tabela D.14 – Exemplos e marcadores superficiais para a relação EVIDENCE

Ordem	Proposição 1	Proposição 2
NS	As arquiteturas sistólicas são as mais adequadas para manipular operações matriciais, pois exploram o paralelismo de granulosidade fina presente nessas operações e possuem um baixo overhead de comunicação e sincronismo.	**Muitos algoritmos bem conhecidos para manipulação de matrizes já foram** mapeados em arquiteturas sistólicas.
	Os Sistemas Computacionais Distribuídos, há mais de uma década, já deixaram de ser apenas uma promessa, não estando mais restritos a pesquisas realizadas nos meios acadêmicos.	Hoje, com um maior ou menor grau de conformidade em relação àquilo que se considera distribuído, **já há** vários exemplos bem-sucedidos de implementações e o número e a diversidade das aplicações distribuídas não pára de crescer, abrangendo ambientes tão diversos como o meio acadêmico, comércio, indústria e residências.
	Muitos artigos já trataram sobre a superexposição de informações ao usuário e a passagem do controle da seqüência do aprendizado do autor para o estudante.	**Experiências reais com alunos têm mostrado** que, na instrução baseada em hipertextos, alguns alunos encontraram dificuldades em ter que tomar muitas

	Logo, a flexibilidade oferecida pode levar o usuário a se perder no hiperdocumento como também dificultar o encontro das informações desejadas.	decisões e de saber navegar dentro de suas lições, fazendo com que se sintam desorientados.

Tabela D.15 – Exemplos e marcadores superficiais para a relação INTERPRETATION

Ordem	Proposição 1	Proposição 2
NS	Devido a pouca capacidade cognitiva, o modelo de apresentação não pode ser alterado, o que faz o sistema agir da mesma forma com todos os estudantes. Uma das necessidades dos CAIs para realmente alcançarem seus objetivos é a flexibilidade no processo de ensino.	**Isto significa** que o sistema precisa ser capaz de deduzir e manter um modelo detalhado do estudante, e utilizar estas informações para individualizar o ensino, modificando a apresentação do conteúdo e a estratégia de ensino, conforme necessário.
	Notadamente, as etapas de entendimento e modificação estão muito relacionadas com a disponibilização das informações do software,	**ou seja,** se apóiam na existência, consistência, completitude e atualização correta dos documentos que o compõem.

Tabela D.16 – Exemplos e marcadores superficiais para a relação JUSTIFY

Ordem	Proposição 1	Proposição 2
NS	Portanto, **é fundamental** a existência de ferramentas de teste que dêem suporte à sua aplicação.	**A disponibilidade de ferramentas de teste contribui para** um desenvolvimento de software de maior qualidade e produtividade. (...)
	um objetivo inicial desta pesquisa se concentra em uma análise sobre a adequação de técnicas formais existentes, como Redes de Petri, Statecharts e CSP (Communicating Sequential Process), em relação a especificação de aplicações hipermídia abertas.	**A escolha das técnicas** formais que serão investigadas neste trabalho de mestrado **se deu** pela grande utilização destas na especificação de sistemas reativos e de tempo real, entre os quais encontram-se as aplicações hipermídia abertas.
	Para a implementação, utilizamos dos recursos do simulador de redes neurais SNNS (Stuttgart Neural Network Software).	**A opção pelo uso** de um software deste tipo **surgiu** pelo fato de não necessitarmos de mudanças quanto à construção dos modelos das redes, ou seja, as redes neurais seriam implementadas exatamente como são propostas nos modelos originais, sendo exatamente o que dispomos no simulador.
	Portanto, os Sistemas Reativos controlam algumas atividades humanas essenciais e por isso, a atividade de teste no desenvolvimento dos mesmos é ainda mais crucial,	**dado que** a ocorrência de falhas nesses sistemas pode colocar em risco vidas humanas ou determinar elevados prejuízos materiais.
	dos quais dois representantes **merecem destaque** no cenário computacional atual: o PVM (Parallel Virtual Machine) e o MPI (Message Passing Interface).	**O PVM destaca-se por** ser considerado por alguns autores um padrão de fato para plataformas de portabilidade, enquanto o MPI é uma tentativa de padronização de direito, levada a cabo por diversas organizações mundiais. (...)

	O Jspell foi construído com base no ispell e estendido no sentido de associar a cada entrada no dicionário um conjunto de atributos como, por exemplo, categoria gramatical, gênero e número. Deste modo, a análise de uma palavra retorna um conjunto de possíveis interpretações da palavra. Isso serve para aplicações em linguagem natural que necessitam de um mecanismo de classificação léxica.	**É um primeiro passo para** implementação de um analisador sintático para a língua.
	Este trabalho tem por objetivo propor um conjunto de requisitos cuja implementação é considerada desejável em sistemas de autoria hipermídia educacional e verificar a abrangência e completude desta proposta através da realização de um estudo de caso. (...)	**Esta proposta é justificada** pela necessidade de se obter um conjunto único de requisitos, que possa suprir as deficiências observadas em trabalhos realizados anteriormente e auxiliar satisfatoriamente as etapas de engenharia de requisitos de novos sistemas e a avaliação de sistemas já existentes.
	Na mesma perspectiva dos estudos de Offutt et al e Wong et al, **este trabalho tem como objetivo** investigar alternativas pragmáticas para a aplicação do critério Análise de Mutantes e, nesse contexto, é proposto um procedimento para a determinação de um conjunto essencial de operadores de mutação para a linguagem C, com base nos operadores implementados na ferramenta Proteum. (...)	**Este trabalho é caracterizado como relevante dado que**: - existe um grande número de programas, para diversas aplicações, desenvolvidos na linguagem C; - tem-se observado que o critério Análise de Mutantes é bastante eficaz em revelar a presença de erros; - o alto custo de aplicação da Análise de Mutantes dificulta sua utilização em ambientes comerciais/industriais; e - a Proteum é a única ferramenta existente atualmente que apóia a aplicação da Análise de Mutantes para o teste de programas escritos em C.
	Apesar de ser uma técnica conhecida há algum tempo, os cenários têm ganhado nos últimos anos grande destaque entre os principais autores na área de desenvolvimento de sistemas.	**Há vários métodos publicados recentemente**, entre eles OMT, Objectory e Fusion, que utilizam a técnica de construção de cenários em suas fases e uma grande quantidade de extensões que utilizam cenários como técnica de apoio.
	Num primeiro passo, para que esse novo modelo seja prático para aplicações reais, há a necessidade de que algumas notações sejam validadas, e que outras sejam reformuladas para adequarem-se melhor ao seu uso cotidiano, mantendo seu embasamento semântico.	**Isso é necessário para** que a notação melhore a representação dos elementos de uma modelagem, gerando diagramas que sejam ao mesmo tempo intuitivos para a compreensão da informação representada, (limpo) não apresente uma densidade demasiadamente elevada de informação em determinados locais e (efetivo) apresente toda a informação necessária ao entendimento do usuário.
	Esse aspecto foi abordado em relação às especificações baseadas em MEFs por Petrenko e Bochmann, que salientam a relevância de pesquisas nessa direção,	**já que** com a análise de cobertura a qualidade da atividade de teste pode ser quantificada.
	Esses problemas não são exclusivos de ensino,	**mas** ocorrem em hiperdocumentos de um modo geral
	Desse modo, uma abordagem sistemática	**Nesse caso, o uso de um modelo, ou de**

	para modelar o conteúdo e definir a organização estrutural é especialmente importante no projeto de aplicações hipermídia grandes e complexas.	**um método, ajuda a** disciplinar a atividade de autoria, para que: - seja possível descrever a aplicação independentemente de sua implementação; desta forma, a definição de um esquema conceitual ajuda a oferecer uma visão mais abstrata do domínio da aplicação do que aquela obtida pela inspeção do seu código, ou pela tentativa de extrair uma semântica das estruturas dos nós e dos elos; - a evolução da aplicação seja mais facilmente fornecida, desde que as decisões de projetos sejam documentadas no nível correto de abstração.
	Muito esforço foi feito na tentativa de se obter etiquetadores cada vez mais precisos para o inglês, como a etiquetagem manual de corpus volumoso, correção da etiquetagem automática também objetivando obter corpus de treinamento maior, desenvolvimento de novas técnicas supervisionadas e não supervisionadas e adaptação de técnicas utilizadas em Aprendizado de Máquina.	**O uso de técnicas de Aprendizado de Máquina se deve** ao fato de etiquetadores poderem ser encarados como classificadores.
	Nesse sentido, a Engenharia Reversa tem por objetivo principal recuperar informações, através da produção de visões do sistema,	**as quais podem facilitar**, primeiramente, o entendimento e, posteriormente, a modificação e revalidação do sistema,
	O paradigma simbolista apresenta vantagens quanto à representação de sentenças com complexidade arbitrária,	**pois** se utiliza das técnicas de parsing, que mapeiam textos de entrada em representações internas, classificando suas constituintes.
	Este trabalho contextualiza-se na área de Engenharia de Software,	**por fornecer** um conjunto de requisitos que auxilia o processo de desenvolvimento de sistemas de uma área específica
	mas essa tarefa é difícil	**porque** os sistemas de autoria hipermídia tradicionais (HyperCard e ToolBook, por exemplo) possuem apenas recursos genéricos para auxiliar o autor no desenvolvimento de aplicações.
	Outro ponto motivador das pesquisas deste trabalho de mestrado foi o de utilizar o método Fusion-RE/I desenvolvido também no ICMC,	**que deve-se**, principalmente, do fato de que a experiência prática conduzida neste projeto se constitui da aplicação da engenharia reversa a um ambiente de hipermídia, e esta aplicação foi registrada no próprio ambiente hipermídia.
	Além disso, foi incluído um estudo das comunicações ponto-a-ponto e coletivas do PVM e do MPI,	**que foram utilizados para** uma melhor avaliação dos resultados obtidos.
	A técnica mais comumente empregada hoje em dia para medir distâncias em robôs móveis que navegam em ambientes fechados é a TOF com ultra-som.	**Sua popularidade se deve** a seu baixo custo e facilidade de utilização.
	A validação do EHDT tem se apresentado	**uma vez que** caracteriza uma fase em que

	como uma atividade especialmente necessária,	se pode questionar, de uma forma mais amadurecida, a eficácia da proposta inicial de seu projeto.
	Os etiquetadores para a língua inglesa atingiram um estado da arte entre 95-99% de precisão geral,	**visto que**, independente da abordagem para etiquetagem escolhida alguns casos acabam não sendo tratados, por exemplo, por dependerem de informações semânticas, o que impõe um limite à precisão geral.
SN	**A utilização de computação concorrente (ou paralela) constitui uma área** de atuação interdisciplinar, **de grande interesse** na atualidade. (...)	A abordagem adotada no Laboratório de Sistemas Digitais (LaSD) do Departamento de Ciências de Computação e Estatística, ICMSC-USP, segue a filosofia de uso compartilhado de uma máquina paralela, baseada na utilização de um banco de transputers, que permite a alocação remota (via rede) e dinâmica de um certo número de transputers, para um usuário do sistema. (...)
	A criação de algoritmos que implementem Redes Neurais Construtivas, chamados Algoritmos Construtivos, é recente nas pesquisas em Redes Neurais Artificiais. (...)	**Da necessidade de** trabalhar com as RNCs em um único ambiente, o Simulador Kipu foi inicialmente projetado. (...)
	Devido a sua grande utilização e características como simplicidade, robustez e eficiência,	o ambiente de passagem de mensagens PVM, desenvolvido por Geist et al, é utilizado como base para o ambiente de passagem de mensagens PVM-W95, responsável pela criação da máquina paralela virtual no Windows95.
	Uma nova representação, mais voltada para a modelagem de situações do mundo real, **é assim de fundamental importância** para que um modelo possa ser aceito e utilizado realmente.	**Este trabalho propõe justamente** estender o SIRIUS neste aspecto, e com isso incrementar sua utilidade prática.
	Com a crescente utilização da informática no contexto educacional, **tornou-se incontestável a necessidade de** desenvolvimento e aprimoramento de tecnologias que pudessem suportar este novo paradigma. (...)	**Logo**, este trabalho foi proposto de forma a colaborar na execução dessas atividades, fornecendo um conjunto de requisitos que possua as características mencionadas anteriormente.
	As várias formas de atuação nesses dois cenários típicos de versionamento para páginas Web mostram que um suporte ao controle de versão dos arquivos para os desenvolvedores, os quais trabalham em um desenvolvimento colaborativo, e um suporte à navegação por versões anteriores das páginas, por parte dos internautas, são alvos de investigação com muito interesse.	**Neste contexto**, **uma ferramenta** que auxilie no trabalho cooperativo entre os desenvolvedores, no gerenciamento das diferentes versões de uma página Web, e forneça um mecanismo para visualização e recuperação da mesma por parte dos internautas **se apresenta como um auxílio de grande utilidade**.
	Considerando que os computadores pessoais conectados por uma rede de comunicação e utilizando o Windows95 possuem potencial suficiente para oferecer os recursos da computação paralela para	**o objetivo deste trabalho é** implementar e descrever detalhadamente uma Máquina Paralela Virtual no Ambiente Windows95.

	um grande número de possíveis usuários,	
	neste contexto, segundo alguns autores, o MPI pode tornar-se um padrão de grande importância no futuro da computação, tanto a nível acadêmico como comercial;	**sob esse ponto de vista, é importante** estudar o comportamento de algumas de suas implementações;
	(...) Sistemas baseados na abordagem léxica apresentaram algumas deficiências como: - ocupavam muito espaço, eram caros e difíceis de implementar e manter; - eram pouco portáteis, se perdendo de acordo com o assunto abordado - um sistema desenvolvido para lidar com diálogos sobre filosofia, por exemplo, seria praticamente inútil para uma base de informações sobre o funcionamento de um computador, embora as estruturas frasais fossem as mesmas; - freqüentemente encontravam problemas com palavras desconhecidas, pois não era possível prever todas as palavras necessárias em um dado domínio; o não reconhecimento de uma palavra, muitas vezes, impossibilitava a continuação da análise da frase.	**Visto isso, é proposta** neste trabalho uma abordagem alternativa. (...)

Tabela D.17 – Exemplos e marcadores superficiais para a relação MEANS

Ordem	Proposição 1	Proposição 2
NS	Para fins de editoração de material didático, o autor necessita de ferramentas que aliem interatividade a mecanismos poderosos de busca.	**A utilização de** índices para responder aos tipos de pesquisa necessários é uma boa alternativa, porém dados de mídia contínua necessitam de novas técnicas de indexação.
	O modelo cliente-servidor representa uma alternativa natural para a implementação da comunicação em sistemas computacionais distribuídos, particularmente aqueles baseados em redes locais de computadores. (...)	**Vários mecanismos podem ser adotados para** a implementação desse modelo. (...)
	Dentre as abordagens existentes, algumas tentam explicitamente modelar a semântica de domínios específicos,	**adotando** estruturas de representação pré-definidas, como o g-IBIS.
	para auxiliar no gerenciamento de versões de páginas Web por meio da própria Web,	**apoiando** os desenvolvedores de uma página no trabalho colaborativo, sem que haja perda ou sobreposição acidental de informações, além de possibilitar aos internautas visualizarem diferentes versões de uma mesma página e localizar as diferenças entre elas.
	A sumarização pode ser vista como o processo de condensar uma fonte de informação (texto-fonte), resultando em uma versão mais curta (sumário), que preserve seu conteúdo informativo.	**Assim, para produzir sumários** deve-se identificar, no texto-fonte, as informações mais relevantes que devem compor o sumário ou, alternativamente, identificar as informações menos relevantes que devem ser omitidas no sumário.

	as quais procuram reduzir a quantidade de mutantes gerados	**através da** redução do número de operadores de mutação utilizados durante o teste.
	Para isso, é necessário especificar a nova classe de tipos de dado áudio,	**o que** no MRO e no GEO **é feito através** da criação de uma nova característica de atributos.
	Hoje, é possível desenvolver rapidamente um projeto de sistema digital	**empregando-se** novas metodologias como linguagens de descrição de hardware (HDLs), ferramentas de síntese lógica e simulação.
	Casos de teste redundantes surgem da tentativa do testador de obter um conjunto de teste que seja adequado ao critério utilizado,	onde **para esse fim** inserem-se diversos casos de teste, sem a preocupação da não redundância.
	Este trabalho tem por objetivo propor um conjunto de requisitos cuja implementação é considerada desejável em sistemas de autoria hipermídia educacional e verificar a abrangência e completude desta proposta através da realização de um estudo de caso.	**Para isso**, foram revisadas diversas publicações relacionadas a este tema, de forma a indicar os requisitos considerados importantes nesses estudos para a implementação de tais sistemas.
	onde a inserção de vídeo é o ponto principal.	**Para tanto**, o professor (autor) necessitará ter acesso privilegiado a um servidor de dados multimídia, com utilitários para busca de vídeos (ou parte de vídeos) que lhe interessem. (...)
	que se comunicam com seus pares	**pela** troca de mensagens utilizando uma linguagem de comunicação.
	Estas limitações procuraram ser contornadas	**por meio de** mecanismos de indexação bastante usados nesta área.
	Este trabalho se propôs a construir um etiquetador simbólico, adaptar para o Português do Brasil três etiquetadores disponíveis via WWW, e combinar os etiquetadores adaptados	**utilizando** técnicas da área de Aprendizado de Máquina.
SN	No entanto, uma questão importante que geralmente não é considerada pela simulação é o aspecto de cobertura da atividade de teste,	**através da qual** é possível a quantificação da qualidade dessa atividade.
	Através de um conjunto de processadores que cooperam e comunicam-se entre si,	grandes problemas são resolvidos mais rapidamente do que se estivessem sendo solucionados por computadores seqüenciais (arquiteturas de von Neumann).
	Usando o MRO,	será definida toda a parte de esquema do banco de dados para inclusão de som.
	Utilizando ferramentas de teste	o conjunto de casos de teste pode ser facilmente obtido para a realização do teste de regressão.

Tabela D.18 – Exemplos e marcadores superficiais para a relação MOTIVATION

Ordem	Proposição 1	Proposição 2
NS	Este trabalho se refere à utilização de técnicas de inteligência artificial como métodos para a fusão de sensores, aplicada na solução de um problema específico da área de robótica. (...)	**A motivação deste trabalho** se deve a um problema bastante comum. Diversas áreas da ciência e engenharia precisam de sistemas que capturem, processem e integrem informações provenientes de várias fontes. (...)
	Dentro deste contexto, este trabalho objetiva fazer um estudo a nível de estrutura e desempenho das rotinas de comunicação ponto-a-ponto do MPI sobre sistemas distribuídos baseados em uma rede de computadores pessoais (PC's) sobre o controle do sistema operacional LINUX (versão do kernel 1.3.20). (...)	**Entre os pontos principais que motivaram a realização deste trabalho**, destacam-se: - As duas principais plataformas de portabilidade na atualidade, são o PVM e o MPI; o PVM está sendo explorado por algumas dissertações de mestrado dentro deste grupo de pesquisa, de maneira que é interessante estudar também o MPI, afim de que se possua um ponto de vista mais abrangente sobre plataformas de portabilidade; - A aceitação da computação paralela está intimamente ligada à possibilidade de portabilidade direta de programas entre sistemas heterogêneos; (...)
SN	Possuindo muitas características opostas às dos sistemas hipermídia, a utilização de alguns recursos tutoriais pode ajudar a contornar alguns destes problemas. Apesar de poderem ser bastante flexíveis e até usarem recursos hipermídia (eles até permitem que o usuário navegue à vontade em módulos especialmente projetados para isso), os sistemas tutores possuem estratégias bem definidas quanto aos roteiros a serem seguidos pelos usuários. Com isto, os estudantes, ao fazerem uso dos sistemas tutores, sentem-se guiados ou acompanhados por alguém.	**Baseado em idéias como estas,** este trabalho estuda uma proposta para a incrementação de um sistema hipermídia com alguns recursos de sistemas tutores. (...)
	Muitos desses sistemas foram feitos baseados em outros sistemas anteriormente por mim desenvolvidos. Por exemplo, com base em um sistema para Controle de Estoque e Emissão de Notas Fiscais de Produtos Agrotóxicos, foi feito um sistema para uma Revendedora de Motocicletas e Peças. Mais tarde, algumas partes desse sistema foram adaptadas para uma Oficina Eletrônica de Reparos. A parte financeira do primeiro sistema por mim construído foi usada quase que integralmente em muitos outros sistemas. As bases de dados também possuem diversos arquivos similares, aproveitando-se toda a idéia de projeto. Nessas atividades de reuso, deparei-me	**Diante disso, surgiu a grande motivação para** estudar uma forma de facilitar o reuso, não somente de trechos de código, mas também de conceitos de análise e projeto, bem como de facilitar a manutenção dos sistemas.

	com todos os problemas citados na seção anterior. (...)	
	A grande revolução que está ocorrendo na área de informática, proveniente de conseqüentes aprimoramentos tanto em hardware como em software, tem contribuído para a sedimentação dos sistemas hipertexto/hipermídia. (...)	**Fatores como estes incentivam** um grande número de pesquisas na área, que procuram desenvolver novas técnicas e características a serem incorporadas aos sistemas hipermídia, buscando também novas áreas de aplicações. (...)
	Segundo Harel, uma das formas de validação de Sistemas Reativos em geral e que conseqüentemente também pode ser utilizada na validação de Redes de Petri é a simulação da especificação. Por causa da complexidade de informações envolvidas, é necessário que a simulação seja apoiada por ferramentas para ser adequadamente realizada. No entanto, uma questão importante que geralmente não é considerada pela simulação é o aspecto de cobertura da atividade de teste, através da qual é possível a quantificação da qualidade dessa atividade.	**Isso motivou** a investigação de outras formas de validação de especificação, como, por exemplo, a Análise de Mutantes. (...)
	Dessa forma, o ideal é a detecção de erros no início do processo,	**o que motiva** o uso de critérios sistemáticos para o teste e validação do sistema ainda na fase de especificação.
	No entanto, as técnicas existentes na indústria tradicional para o desenvolvimento de sistemas não satisfazem os requisitos das aplicações hipermídia,	**motivando** a pesquisa de novos modelos e métodos que forneçam diretrizes para gerenciar de maneira sistemática o projeto e o desenvolvimento desse tipo de aplicação

Tabela D.19 – Exemplos e marcadores superficiais para a relação OTHERWISE

Ordem	Proposição 1	Proposição 2
NS	deve-se identificar, no texto-fonte, as informações mais relevantes que devem compor o sumário	**ou, alternativamente,** identificar as informações menos relevantes que devem ser omitidas no sumário.
	e, caso a resposta seja positiva, obter mais um argumento em favor da qualidade do SASHE.	**Caso contrário**, o experimento teria o valor de identificar as deficiências relacionadas com a implementação desses requisitos iniciais.

Tabela D.20 – Exemplos e marcadores superficiais para a relação PURPOSE

Ordem	Proposição 1	Proposição 2
NS	Nessa técnica, a característica mais importante é examinar como um critério de teste pode ser utilizado no teste de regressão	**a fim de** ajudar os testadores a determinar quais casos de teste devem ser selecionados ou ter uma maior prioridade para o processo de revalidação das novas funcionalidades.
	algumas abordagens vêm sendo propostas,	**as quais procuram** reduzir a quantidade de mutantes gerados através da redução do número de operadores de mutação utilizados durante o teste.

	Nesta dissertação, também avalia-se a representação dos construtores semânticos adotada em diferentes modelos de dados semânticos, orientados a objetos descritos na literatura,	**buscando-se** uma representação adequada para os conceitos de SIRIUS.
	Segundo Myers, teste é um procedimento de executar um programa	**com a intenção de** encontrar erros existentes.
	Muitas empresas têm incorporado tecnologia orientada a objetos no seu processo de desenvolvimento de software	**com a perspectiva de** aprimorar a qualidade dos produtos de software.
	Foi selecionado como domínio para um protótipo de aplicação do SIATE o tópico Aquisição de Conhecimento para Sistemas Inteligentes, onde são abrangidos os subtópicos de Aquisição de Conhecimento Explícito e Implícito. Esta seleção se deve ao gargalo existente no processo de aquisição de conhecimento especialista.	**Com isso, a função principal** dessa aplicação do SIATE é auxiliar engenheiros de conhecimento, público alvo do ambiente, em aprender como efetuar aquisição de conhecimento de forma correta e eficiente.
	O objetivo não é tentar impedir iniciativas do aprendiz, mas, sim, fornecer recursos ao autor para que critérios como relevância e fidelidade às metas de ensino/aprendizado sejam consideradas.	**Com isso, procura-se** minimizar os riscos de o aprendiz se perder por caminhos irrelevantes.
	Com a crescente demanda de software e a conseqüente evolução da Engenharia de Software, atividades agregadas sob o nome de Garantia de Qualidade de Software têm sido introduzidas ao longo de todo o processo de desenvolvimento, entre elas as atividades de VV&T (Verificação, Validação e Teste),	**com o intuito de** auxiliar na melhoria da qualidade e da produtividade.
	O GEO tem sido construído no Instituto de Ciências Matemáticas de São Carlos	**com o objetivo de** validação prática e estudo de como os conceitos de orientação a objetos podem ser implementados em um sistema de software real.
	Foi construída também uma interface, permitindo a manipulação do áudio música no Gerenciador de Objetos,	**com o propósito de** permitir avaliar a implementação desse suporte.
	Esta dissertação é o resultado de um trabalho	**cujo objetivo inicial é** investigar a aplicação de Redes Neurais Construtivas, RNCs, em tarefas de Reconhecimento de Padrões.
	Para isso, foram revisadas diversas publicações relacionadas a este tema,	**de forma a** indicar os requisitos considerados importantes nesses estudos para a implementação de tais sistemas.
	A utilização do SASHE foi motivada pela constatação da necessidade de se obter informações sobre as condições atuais de sua implementação,	**de forma que** os resultados obtidos pudessem contribuir para a evolução do projeto.
	Por fim, será analisado o comportamento das três implementações face a uma aplicação paralela real,	**de maneira a** comparar-se os resultados obtidos com situações reais de paralelismo.
	A notação originalmente proposta teve como objetivo descrever as modelagens	**de maneira que** ficasse bem claro o embasamento teórico que suporta SIRIUS.

	efetuadas,	
	A disponibilidade de ferramentas de teste oferece também recursos para o desenvolvimento de estudos empíricos	**de modo a** avaliar o custo e a eficácia das técnicas e critérios de teste nos quais as ferramentas se baseiam.
	sendo que pontos podem ser definidos	**de modo que** a ferramenta execute determinadas ações quando ocorrem situações específicas;
	Nessa dissertação são discutidos o projeto e a implementação de um módulo de simulação de redes de computadores para o ASiA. (...)	**Desta forma, o objetivo deste trabalho é** estender o ASiA para que este ofereça os recursos necessários para a simulação de redes de computadores e implementar uma versão inicial do módulo de redes que poderá ser facilmente estendido, através da inserção de novos modelos.
	Este trabalho está em consonância com os demais trabalhos desenvolvidos pelo Grupo de Engenharia de Software do ICMC/USP	**e objetiva** contribuir na determinação de formas alternativas e de baixo custo para a aplicação do critério Análise de Mutantes.
	Nesse contexto, esse projeto de pesquisa propõe a validação do Educational Hyperdocuments Design Tool (EHDT),	**especialmente desenvolvido para** possibilitar a modelagem de hiperdocumentos para o Sistema de Autoria e Suporte Hipermídia para Ensino (SASHE)
	Deste modo, a análise de uma palavra retorna um conjunto de possíveis interpretações da palavra.	**Isso serve para** aplicações em linguagem natural que necessitam de um mecanismo de classificação léxica.
	que possibilite ao autor a criação de roteiros	**na intenção de** oferecer guided tours a seus usuários,
	Mais recentemente, o Grupo de Engenharia de Software do ICMC-USP tem desenvolvido pesquisas sobre teste de software orientado a objetos,	**na tentativa de** definir algumas estratégias para esse tipo de teste e de implementar ferramentas que automatizem o teste OO.
	Este trabalho propõe uma abordagem alternativa à usual, que é descartar o código antigo quando se cogita de ampliar a funcionalidade de um sistema existente, com idade avançada e desatualizado.	**O objetivo dessa abordagem é** reconhecer padrões de software, sejam eles de análise, de projeto ou de código, que sejam úteis no reuso, na ampliação de funcionalidade e na manutenção de sistemas.
	AT&T Internet Difference Engine (AIDE) é um sistema que fornece algum suporte de versão usando o sistema RCS.	**O objetivo principal do AIDE é** permitir que os usuários (internautas e/ou desenvolvedores) vejam as diferenças entre páginas Web quando elas são atualizadas.
	O sistema SASHE foi desenvolvido por equipe de pesquisadores do Instituto de Ciências Matemáticas e de Computação de São Carlos, sob orientação da Professora Dra. Maria das Graças Volpe Nunes, no âmbito do projeto ProTeM-CC-FaseIII, HyperProp.	**O projeto teve como objetivo** implementar um sistema que explorasse as potencialidades do Modelo de Contextos Aninhados (MCA), descrito posteriormente nessa monografia, e que servisse de ferramenta para usuários da comunidade universitária no âmbito do processo de ensino e aprendizagem.
	Nesse sentido, diversos experimentos empíricos têm sido desenvolvidos	**objetivando** buscar formas alternativas para viabilizar a aplicação desse critério, assim como comparar o custo, strength e a eficácia deste com outros critérios.
	A grande exigência dos clientes por melhores softwares tem obrigado os	**para** continuarem competindo no mercado.

	desenvolvedores a aperfeiçoarem o seu produto final	
	Desta forma, o objetivo deste trabalho é estender o ASiA	**para que** este ofereça os recursos necessários para a simulação de redes de computadores
	Este trabalho se propôs a construir um etiquetador simbólico, adaptar para o Português do Brasil três etiquetadores disponíveis via WWW, e combinar os etiquetadores adaptados utilizando técnicas da área de Aprendizado de Máquina.	**Pretendeu-se**, assim, desenvolver um trabalho comparativo bastante extenso para a escolha de um etiquetador que etiquete com melhor precisão uma gama variada de tipos de texto em português do Brasil.
	o objetivo deste trabalho é o de complementar o ambiente de ensino (estudante) com funções diretas, não inseridas dentro das ligações (links),	**procurando**: aprimorar a interação com o estudante, facilitar em situações de dificuldade ou dúvida, ajudar na procura de informações complementares, etc.
	O Grupo de Engenharia de Software do Instituto de Ciências Matemáticas e de Computação - ICMC/USP - vem desenvolvendo atividades de pesquisa concentradas no estudo de princípios, estratégias, métodos e critérios de teste e validação na produção de software, assim como na especificação e implementação de ferramentas	**que apóiem** a realização das atividades de teste e viabilizem a avaliação do aspecto complementar desses critérios, através de estudos empíricos.
	A engenharia de requisitos, de uma forma geral, lida com o problema de captar as necessidades dos vários usuários de um sistema e traduzir essas necessidades na forma de requisitos,	**que podem ser usados para** medir uma eventual implementação.
	Existem algumas estratégias propostas	**que visam** a obtenção do menor conjunto possível a partir de um conjunto inicial,
	Segundo Fortes, ambos modelos têm em comum a identificação de uma camada central,	**responsável pela** "resolução" das interligações entre as unidades de informação, cujo conteúdo se encontra disponível pelos mecanismos convencionais nos dispositivos de memória dos computadores.
	Buscando reduzir os custos associados com essa atividade, técnicas e critérios de teste são propostos,	**servindo para** conduzir e avaliar a qualidade da atividade de teste.
	Seu desenvolvimento utiliza, como ponto de partida, um metamodelo, que permite representar, de maneira uniforme, os elementos essenciais de qualquer modelo de dados orientado a objetos,	**tendo como objetivo** atender às necessidades de suporte à construção de software para ambientes de projetos de engenharia.
	Com esse objetivo, muitas técnicas vêm sendo criadas	**visando** à produção automática de sumários
SN	**Para que** o processo de software possa cumprir seus objetivos	é necessário um planejamento detalhado que mostre a realidade do processo atual, a meta que se almeja com a melhoria, a estratégia para se atingir essa meta e os planos de ação.
	Com o objetivo de analisar automaticamente a estrutura de frases	existem vários estudos sobre o processamento da língua natural.

	Para amenizar os problemas desses usuários	foi desenvolvido um ambiente modular de auxílio e ensino de escrita técnica chamado AMADEUS (Amiable Article Development for User Support).
	A fim de solucionar este problema,	o World Wide Web Consortium (W3C) elaborou recentemente a recomendação SMIL – Synchronized Multimedia Integration Language – que provê uma linguagem para especificação de sincronização de objetos multimídia no contexto da WWW.
	Buscando reduzir os custos associados com essa atividade,	técnicas e critérios de teste são propostos
	Com o intuito de aperfeiçoar o uso de informática na educação,	os CAIs são alvo de várias pesquisas sobre como melhorar a performance educacional em ambientes voltados para o ensino.
	Na tentativa de reduzir o custo da atividade de teste,	várias técnicas e critérios têm sido propostos para auxiliar sua condução e avaliação.
	Procurando estender a aplicação do critério Análise de Mutantes para o teste de integração,	um novo critério denominado Mutação de Interface (Interface Mutation-IM) foi desenvolvido.
	Visando a facilitar a especificação do aspecto comportamental de Sistemas Reativos,	várias técnicas de especificação foram propostas.

Tabela D.21 – Exemplos e marcadores superficiais para a relação RESTATEMENT

Ordem	Proposição 1	Proposição 2
NS	Para que um usuário possa elaborar e/ou utilizar uma modelagem, é necessário que essa representação gráfica seja clara, limpa, concisa e efetiva. Por outro lado, deve permitir que os conceitos de modelagem do modelo em si sejam também claramente expressos. Este último objetivo foi atendido pela representação de SIRIUS proposta em seu documento original, que privilegiou a manutenção da representação explícita dos conceitos de modelagem nos diagramas que representam as modelagens do mundo real efetuadas com o modelo. No entanto, em situações reais de modelagem, esse critério deve estar subordinado, em primeiro lugar, à clareza e efetividade da informação representada.	**Assim**, para que este modelo possa ser uma ferramenta adequadamente utilizável por projetistas, analístas e programadores de gerenciadores de Bases de Dados, é necessário que a notação diagramática do modelo priorize a intuitividade e a clareza das modelagens efetivadas, e não apenas o compromisso com a teoria que embasa o modelo.
	Com o objetivo de representar a base de dados de uma forma mais compacta, as palavras do dicionário são armazenadas sem prefixos e sufixos e apresentam flags associados que servem para indicar a existência desses. Isto evita a necessidade de armazenar todas as palavras com afixos no dicionário.	**Desta forma**, cada item do vocabulário é um radical associado a flags, que representam os possíveis prefixos e sufixos que podem ser utilizados para derivar palavras daquele radical. (...)
	Um dos problemas que surgiu com o	**Isto é**, como avaliar o funcionamento e

	constante avanço de arquiteturas é a forma de avaliar as novas abordagens que vêm sendo propostas.	principalmente o desempenho das arquiteturas propostas.
	O ponto forte desse tipo de sistema é poder apresentar ao estudante o caminho percorrido,	**ou seja**, os mecanismos que foram utilizados para se alcançar a resposta.

Tabela D.22 – Exemplos e marcadores superficiais para a relação RESULT

Ordem	Proposição 1	Proposição 2
NS	O controle que o estudante tem possibilita-o a realizar escolhas	**afetando** o andamento do aprendizado
	Assim, uma ferramenta que automatize a fase de engenharia de requisitos (baseada em uma técnica completa e relativamente simples) capaz de captar, modelar e validar requisitos, pode auxiliar em muito na especificação de um dado sistema,	**agilizando** o seu processo de desenvolvimento.
	Entretanto, as regras devem ser explicitamente programadas, tendo em mente exemplos específicos.	**Aí surgem** problemas quanto à confecção das regras gramaticais do português. (...)
	Ambos são modelados a partir da teoria dos grafos e segundo o paradigma da orientação a objetos.	**Assim**, software e hardware são reduzidos a um grafo e o modelo global produzido é totalmente orientado a objetos.
	Apesar de poderem ser bastante flexíveis e até usarem recursos hipermídia (eles até permitem que o usuário navegue à vontade em módulos especialmente projetados para isso), os sistemas tutores possuem estratégias bem definidas quanto aos roteiros a serem seguidos pelos usuários.	**Com isto**, os estudantes, ao fazerem uso dos sistemas tutores, sentem-se guiados ou acompanhados por alguém.
	Buscando uma alternativa para reduzir a distância entre essas duas estratégias, chega-se à hipermídia adaptativa que, através da definição de um conjunto de mecanismos de navegação (Adaptative Navigation Support - ANS), apresenta aos usuários roteiros que sejam potencialmente de seu interesse.	**Conseqüentemente**, essa classe de sistemas deve ser capaz de estabelecer um modelo de usuário, garantindo-se, assim, a eficácia de seus hiperdocumentos, ao evitar que o usuário navegue superficialmente pelos relacionamentos dos mesmos.
	Atualmente, as duas áreas têm convergido,	**de maneira que** a combinação entre os dois enfoques computacionais oferece benefícios para ambos os lados.
	A UML é uma linguagem de modelagem orientada a objetos que não estabelece um padrão de processo de desenvolvimento.	**Dessa forma**, organizações e projetos que se enquadram em diferentes processos de desenvolvimento podem utilizar a mesma notação para os seus modelos.
	Uma rede mínima possui apenas camadas de entrada e saída, necessárias para a correta representação do conjunto de dados utilizado no treinamento da Rede Neural. (...)	**Desta maneira**, uma topologia é criada de acordo com a necessidade do problema abordado, dispensando uma busca exaustiva pela topologia mais adequada.
	O Jspell foi construído com base no ispell e estendido no sentido de associar a cada entrada no dicionário um conjunto de	**Deste modo**, a análise de uma palavra retorna um conjunto de possíveis interpretações da palavra. (...)

	atributos como, por exemplo, categoria gramatical, gênero e número.	
	Nesse ambiente, o usuário pode simplesmente navegar pelo hiperdocumento, como também remodelar a sua estrutura organizacional de acordo com os recursos oferecidos pelo ambiente,	**e com isso** obter um material com as características adicionais do SASHE.
	Além disso, existem superfícies que absorvem a radiação do SONAR,	**fazendo com que** esse tipo de sensor não as detecte.
	O desenvolvimento de computadores paralelos envolve uma grande quantidade de recursos,	**gerando** produtos finais caros e normalmente deixando o usuário final dependente apenas de um fabricante.
	Nos últimos anos, com o avanço cada vez mais rápido dos computadores e da tecnologia relacionada, esta explosão de informações está cada vez mais intensa. Diariamente, trilhões de unidades de informação circulam pelo mundo.	**Graças a este avanço**, textos em formato eletrônico são facilmente obtidos - textos clássicos de grandes escritores, textos de jornais, publicações científicas, etc. - com milhões de palavras e estruturas lingüísticas das mais variadas.
	A computação paralela em ambientes paralelos virtuais é bastante versátil, permitindo que a busca por alto desempenho possa ser exercitada a partir de plataformas computacionais distribuídas.	**Isso traz** benefícios diversificados, incluindo uma possível queda na relação custo/benefício da computação paralela, uma vez que ambientes distribuídos (por exemplo, redes de estações de trabalho ou computadores pessoais) podem ser utilizados para esse fim.
	Muitos artigos já trataram sobre a superexposição de informações ao usuário e a passagem do controle da seqüência do aprendizado do autor para o estudante.	**Logo**, a flexibilidade oferecida pode levar o usuário a se perder no hiperdocumento como também dificultar o encontro das informações desejadas.
	pois o usuário (leitor/aprendiz) tende a perder o sentido de localização e direção das informações (desorientação) à medida que navega de uma página para outra em uma estrutura rasa, ou não-hierárquica, de relacionamentos.	**Nesse caso, o usuário acaba** desprendendo um esforço adicional de concentração (sobrecarga cognitiva).
	Após o surgimento de técnicas e métodos sistemáticos, especialmente elaborados para apoiar o desenvolvimento de software, várias alterações e melhorias foram propostas para essa atividade,	**o que causou** uma grande revolução na maneira segundo a qual o software era criado.
	As primeiras implementações, resultantes do surgimento do elemento software, em contrapartida ao elemento hardware, eram realizadas sem qualquer tipo de administração,	**o que resultava**, na maioria das vezes, em prazos esgotados e em custos elevados.
	há uma necessidade de se construir um número grande de regras para se cobrirem todos os casos,	**o que torna** o sistema cada vez mais complexo.
	e, em alguns casos, são abrangentes o suficiente para serem lidos no lugar do texto original,	**permitindo** o acesso a mais informações em menos tempo.
	Através desta definição, será possível o armazenamento de som no MRO, permitindo a recuperação do som desejado.	**Por isso**, a busca deverá ser feita através da partitura e instrumentos, que estarão ligados ao som. As músicas serão

	É importante esclarecer que, independente dos atributos criados, não será possível a recuperação de som através da sua reprodução. Por exemplo, considerando que esteja gravada a música Eu sei que vou te amar - Tom Jobim - esta não poderá ser recuperada pela gravação da música em PCM - Pulse Code Modulation - mesmo que seja com a voz de Tom Jobim, pois é muito difícil que as ondas sonoras gravadas na primeira vez sejam idênticas a uma outra.	recuperadas de acordo com suas partituras, enquanto os instrumentos serão relacionados a seus parâmetros.
	Cada modelo de RNC possui seu próprio critério de inserção de novos neurônios e conexões. A diferença entre estes critérios permite a construção de redes diferentes para um mesmo problema.	**Portanto**, não é possível determinar qual modelo é mais propício para um determinado problema sem um estudo comparativo do desempenho das RNCs.
	Esta escolha torna-se então um problema empírico,	**resultando** em perda de tempo e esforço na busca pela topologia mais adequada para o problema abordado.
	A linguagem HTML se consolidou na WWW devido à sua simplicidade,	**tendo se tornado** um padrão de facto.
	O custo desse sistema incluindo o custo do equipamento, da manutenção e do piloto é elevado,	**tornando** seu uso impraticável na maioria das propriedades rurais.
SN	Segundo Pressman, a fase de manutenção consome aproximadamente 60% do orçamento de software de uma organização de desenvolvimento.	**Dentre as várias causas** citadas para esse custo elevado estão as negligências cometidas durante a definição, projeto, codificação e teste do programa. Uma atividade de teste mal conduzida aumenta o custo de manutenção devido a muitas mudanças de correção que precisam ser feitas no software quando os erros forem detectados pelo usuário.
	Eles são limitados pedagogicamente	**porque** sempre utilizam a estratégia estímulo-resposta.
	No ambiente WWW, os internautas freqüentemente se surpreendem ao visitar uma página e percebem que esta já não possui o mesmo conteúdo ou até mesmo que ela não existe mais;	**tudo isso é decorrente** da rápida e natural evolução das informações na WWW.
	Os dados são consultados/alimentados nas bases de dados de organizações independentes entre si, através de alguma forma de intervenção manual,	**uma vez que** a não existência de um esquema comum impede que os dados de uma base possam ser intercambiados com os de outra base.

Tabela D.23 – Exemplos e marcadores superficiais para a relação SOLUTIONHOOD

Ordem	Proposição 1	Proposição 2
SN	A grande evolução dos computadores nas últimas décadas vem sendo acompanhada de perto pelo considerável aumento, tanto em número quanto em complexidade, das aplicações que utilizam estas máquinas. Desta forma, por mais que as arquiteturas tenham apresentado um processo evolutivo intenso, **ainda é necessário buscar um melhor desempenho** para a execução das aplicações que crescem tanto em volume como em complexidade muito mais rapidamente que o hardware disponível.	**Nesse contexto surge** a computação paralela que tem como objetivo principal o aumento de desempenho na execução de uma aplicação. Nessa busca por melhor desempenho os sistemas paralelos apresentam-se como uma alternativa à máquina de von Neumann, sejam esses sistemas paralelos virtuais (sistemas distribuídos) ou em arquiteturas realmente paralelas (máquinas com múltiplos processadores).
	Entretanto, seu **alto custo** de aplicação, decorrente principalmente do grande número de mutantes gerados, tem motivado a proposição de diversas alternativas para a sua aplicação.	**As soluções mais exploradas são** as que procuram diminuir o número de mutantes a serem executados e analisados; os resultados obtidos demonstram que é possível reduzir sensivelmente o número de mutantes gerados sem comprometer a **eficácia** do critério em revelar a presença de erros.
	Na fotografia aérea convencional são utilizados aviões comerciais de 2 ou 4 lugares com baixa velocidade de vôo e equipamentos convencionais de fotografia. O **custo** desse sistema incluindo o custo do equipamento, da manutenção e do piloto **é elevado**, tornando seu uso **impraticável** na maioria das propriedades rurais.	**Entretanto, o sistema descrito pode ser substituído** por um outro baseado em uma aeronave rádio controlada. (...)
	Um dos entraves que se encontra na área da programação paralela corresponde ao forte acoplamento existente entre o software em desenvolvimento e o hardware alvo, isto é, o software é, normalmente, fortemente dependente do hardware e o desenvolvimento visando bom desempenho (um dos objetivos fundamentais da computação paralela) leva à **necessidade de se adequar**, de modo coerente, o software ao hardware. Outro **fator preocupante** nesta área ainda é o **custo**.	**Na tentativa de solucionar** essa situação, apareceram os sistemas distribuídos, que tiveram como objetivo inicial o compartilhamento de recursos. (...)
	Entretanto, quando roteiros são utilizados em um sistema hipermídia, surge o **dilema** entre satisfazer os objetivos do autor do hiperdocumento ou preservar a liberdade do usuário.	**Buscando uma alternativa** para reduzir a distância entre essas duas estratégias, chega-se à hipermídia adaptativa que, através da definição de um conjunto de mecanismos de navegação (Adaptative Navigation Support - ANS), apresenta aos usuários roteiros que sejam potencialmente de seu interesse. (...)
	As aplicações não convencionais, em geral aproveitam os SGBDs já existentes, mesmo que estes **não sejam adequados** para estas aplicações. Esse aspecto faz que a	**Para sanar os problemas** do emprego dos SGBDs disponíveis em aplicações não convencionais, muitas pesquisas estão sendo realizadas, para que novos modelos

	falta de um núcleo comum de gerenciamento de dados **traga problemas** de compatibilidade entre os sistemas.	de dados possam dar o suporte necessário a uma ampla faixa de aplicações não convencionais. (...)
	Esta generalidade, apesar de ser de grande contribuição para sua utilidade e eficiência, faz com que o ambiente apresente algumas **limitações** pela **falta** de conhecimento do domínio.	**Estas limitações procuraram ser contornadas** por meio de mecanismos de indexação bastante usados nesta área.
	No entanto, para certas aplicações, a arquitetura de von Neumann **não satisfaz** os requisitos de desempenho almejados, além de apresentar um **custo muito elevado**, quando supercomputadores com características avançadas são considerados.	**Uma opção para sobrepor esse problema é** a utilização de arquiteturas paralelas. Nesse nível, tem-se a utilização de multiprocessadores ou multicomputadores, que permitem a realização de mais de uma instrução paralelamente.
	No entanto, Castro, por sua vez, questiona que buscas em catálogos de informações mostram a **dificuldade** em encontrar sites com informações científicas realmente interessantes. Muitas vezes, a **decepção** vem na forma de páginas com ênfase no tratamento gráfico e informações **pouco relevantes**, **escassas** ou **desatualizadas**. Nesse caso, o aprendiz pode acabar baseando suas pesquisas em informações que não passaram pelo crivo de um professor, **arriscando-se a propagar erros** científicos.	**Uma solução intuitiva para esse problema pode ser** encontrada em Costa e Xexéo, onde o professor, além de fornecer suporte tecnológico, fornece endereços iniciais de pesquisa e verifica se as informações consultadas pelos aprendizes são compatíveis com os objetivos de ensino almejados. (...)
	Essa atividade apresenta-se bastante **onerosa** podendo, em alguns casos, consumir 40% dos **custos** de desenvolvimento do software.	**Buscando reduzir os custos** associados com essa atividade, técnicas e critérios de teste são propostos, servindo para conduzir e avaliar a qualidade da atividade de teste.
	Um dos fatores **preocupantes** na área de computação paralela, ainda é o seu **custo**. O desenvolvimento de computadores paralelos envolve uma grande quantidade de recursos, gerando produtos finais **caros** e normalmente deixando o usuário final **dependente** apenas de um fabricante.	Atualmente **propostas alternativas buscam** justamente inverter essa situação.
	Um **problema** relacionado à computação paralela é o **alto custo** de aquisição e manutenção de arquiteturas paralelas. Além disso, a compra de uma arquitetura geralmente implica na **dependência** do comprador ao fabricante.	**Uma tendência atual, neste contexto,** é a utilização de sistemas distribuídos como plataformas de execução paralela, a fim de que se forneça **menor custo** de implantação e **maior flexibilidade** no processo computacional paralelo.
	Esta escolha torna-se então um **problema** empírico, resultando em **perda de tempo e esforço** na busca pela topologia mais adequada para o problema abordado.	**Com o objetivo de suprir esta limitação**, as Redes Neurais Construtivas (RNCs) oferecem uma abordagem atrativa para a construção incremental de topologias a partir de uma rede mínima.

Tabela D.24 – Exemplos e marcadores superficiais para a relação SUMMARY

Ordem	Proposição 1	Proposição 2
NS	Neste contexto, enquadra-se o SIRIUS, um Modelo de Dados Orientado a Objetos baseado em Abstrações de Dados, o qual está sendo desenvolvido pelo Grupo de Base de Dados e Imagens do ICMC. Seu desenvolvimento utiliza, como ponto de partida, um metamodelo, que permite representar, de maneira uniforme, os elementos essenciais de qualquer modelo de dados orientado a objetos, tendo como objetivo atender às necessidades de suporte à construção de software para ambientes de projetos de engenharia. Uma característica do modelo é que ele presta-se bem à implementação de um Gerenciador de Objetos que o suporte, sendo que seu formalismo apresenta indicações de como essa implementação deve ser realizada, o que oferece uma indicação também formal de como uma modelagem feita em SIRIUS pode ser mapeada para implementação em um outro modelo. Uma implementação de um Gerenciador de Objetos que suporta SIRIUS, denominado SIRIUS/GO, está sendo desenvolvida pelo Grupo de Base de Dados e Imagens do ICMC. SIRIUS inclui, entre outros, os conceitos de: objetos, atributos, características de atributos, tipos de atributos e tipos de objetos. Tais conceitos são organizados semanticamente e sintaticamente, utilizando-se três abstrações fundamentais: abstração de classificação, abstração de associação e abstração de generalização, das quais a segunda ocorre especializada em duas outras que são: abstração de composição e abstração de agregação.	**Resumindo**, SIRIUS é um modelo que permite a construção de sistemas que representam, de maneira uniforme, os elementos de qualquer modelo de dados orientado a objetos, e em particular, atende às necessidades de ambientes para apoio ao projeto de engenharia e aplicações científicas.
	para corrigir as imprecisões de cada um, melhorar a repetibilidade das medições e estender a capacidade de medição para distâncias e ângulos em que as medições dos sensores são, individualmente, ruins,	**ou seja**, em distâncias e ângulos para os quais nenhum dos sensores são adequados.

Tabela D.25 – Exemplos e marcadores superficiais para a relação CONTRAST

Ordem	Proposição 1	Proposição 2
NN	A visão de que o uso de computadores na educação promove o aumento da produtividade, ensinando mais a mais gente em menos tempo, é o que baseava inicialmente as investigações sobre o uso de informática na educação.	**Atualmente**, o que se enfatiza é o provável potencial oferecido pelo uso do computador no desenvolvimento de habilidades cognitivas que proporcionam melhores meios de ensino e aprendizagem.
	Um concordanceador é um programa que recupera todas as ocorrências de uma determinada cadeia de caracteres em um corpus e as listam, permitindo inclusive que estas listas sejam manipuladas.	**Contudo**, certos tipos de análise não podem ser obtidos apenas através da grafia das palavras sem a utilização de outras características das palavras em questão, como por exemplo, informações de natureza gramatical.
	As expectativas de qualidade de produtos de software têm aumentado cada vez mais,	**enquanto** restrições de custo e recursos humanos estão diminuindo.
	A técnica estrutural estabelece os requisitos a partir da implementação do programa,	**enquanto que** a técnica baseada em erros estabelece os requisitos de teste a partir de erros típicos cometidos no processo de desenvolvimento de software.
	Segundo Pressman, a atividade de teste de software é um elemento crítico para a garantia de qualidade de software e representa a última revisão da especificação, projeto e codificação.	**Entretanto**, a atividade de teste é uma das mais onerosas do desenvolvimento de software.
	Obter esse resultado corresponde a introduzir estratégias pedagógicas nas aplicações hipermídia,	**mas** essa tarefa é difícil porque os sistemas de autoria hipermídia tradicionais (HyperCard e ToolBook, por exemplo) possuem apenas recursos genéricos para auxiliar o autor no desenvolvimento de aplicações.
	que podem ser organizados em um conjunto relativamente pequeno de estruturas,	**mas por outro lado**, com um grande volume de dados estruturalmente idênticos.
	A arbitrariedade na definição de links entre os nós permite grande flexibilidade	**mas, em contrapartida**, tem muitas vezes, como resultado, um hiperdocumento no qual os usuários facilmente se tornam desorientados
	Ao se melhorar a qualidade do processo de software, tem-se maior probabilidade de se obter um produto final mais adequado às expectativas do cliente,	**no entanto**, a realização de uma melhoria do processo de software não é uma tarefa trivial.
	um adjetivo regular em português possui 4 formas distintas (masculino, feminino, singular, plural),	**o que contrasta com** uma única forma do inglês
	Eles são limitados pedagogicamente porque sempre utilizam a estratégia estímulo-resposta.	**Por outro lado**, podem ser eficazes para atingir o objetivo proposto: treinamento.
	O método utilizado para armazenamento do vocabulário é bastante compacto,	**porém** apresenta limitações para aplicações do tipo do Jspell, que necessita guardar informações adicionais para cada palavra.
	Enquanto na primeira versão qualquer	a nova versão permite que novas estruturas

	extensão ao projeto era dependente do modelo de dados implementado e deveria adaptar-se à estrutura já definida pelo módulo principal do programa,	de dados sejam facilmente acopladas ao módulo principal, reutilizando partes (aqui chamadas de componentes) já implementadas ou criando novos componentes necessários para a correta execução do treinamento de Redes Neurais. (...)

Tabela D.26 – Exemplos e marcadores superficiais para a relação LIST

Ordem	Proposição 1	Proposição 2
NN	Assim, deve-se preservar a independência do gerenciamento e da estrutura das bases de uma organização,	**ao mesmo tempo** em que é importante propiciar meios para que as bases de dados de duas organizações independentes possam compartilhar informações.
	Com a Verificação busca-se garantir que o produto está sendo desenvolvido da forma correta,	**ao passo que** com a Validação tenta-se se assegurar que o produto que está sendo desenvolvido está em conformidade com as expectativas do cliente.
	Coad apresenta padrões de análise bem gerais, úteis principalmente no desenvolvimento de sistemas de informação na área empresarial.	**Gamma apresenta** padrões de projeto de utilização bastante ampla que abrange sistemas os mais diversos, para várias áreas de aplicação.
	A atividade de teste da evolução do sistema é geralmente denominada de teste de regressão e visa a fornecer evidências de que as mudanças ocorridas não afetam adversamente as características previamente existentes,	**assim como** a evidenciar que as eventuais novas funcionalidades estão de acordo com as expectativas do usuário,
	O objetivo desses estudos é medir o desempenho do critério Análise de Mutantes,	**bem como** verificar qual o seu relacionamento com outros critérios visando ao estabelecimento de uma estratégia de teste de baixo custo e alta eficácia em revelar a presença de erros.
	Já no início da década de 90, **Nielsen e Streitz sugeriram** a exploração de ambientes cooperativos suportados por hiperdocumentos para facilitar a organização e coordenação de idéias.	**Streitz comentava** que essa união resultaria em sistemas hipermídia multi-usuários distribuídos, ao passo que as atividades de cooperação seriam beneficiadas com o suporte a documentos estruturados.
	tanto para auxiliar a geração de conjuntos de casos de teste	**como para** auxiliar a avaliação da adequação de conjuntos de casos de teste.
	O autor tanto pode desenvolver todo o material que irá compor o hiperdocumento,	**como também** pode partir de um documento já pronto. Este documento pode até ter sido construído por outra pessoa e desenvolvido com o pensamento explícito de servir como um repositório de objetos hipermídia.
	Segundo Fortes, ambos modelos têm em comum a identificação de uma camada central, responsável pela "resolução" das interligações entre as unidades de informação, cujo conteúdo se encontra disponível pelos mecanismos convencionais	**De acordo com Osterbye e Wiil**, a tendência atual no projeto de sistemas hipermídia é o desenvolvimento de sistemas que sejam abertos, extensíveis e distribuídos entre diferentes usuários.

	nos dispositivos de memória dos computadores.	
	Do ponto de vista teórico, procuram-se estabelecer propriedades e características dos critérios de teste, tais como sua complexidade (número máximo de casos de teste requeridos no pior caso) ou uma relação de hierarquia entre os mesmos.	**Do ponto de vista** empírico, dados e estatísticas são coletados, registrando, por exemplo, a freqüência com que diferentes estratégias de teste revelam a presença de erros em um determinado conjunto de programas.
	A verificação é composta de atividades que asseguram a construção correta do software,	**e** a validação envolve atividades que procuram garantir que o software construído atenda aos requisitos do usuário.
	A primeira visa o auxílio à autoria por parte de um autor (professor) de roteiros pré-planejados, que servirão de guias para o aprendizado de um determinado domínio.	**E a segunda**, no auxílio à navegação por parte de um usuário (estudante), na intenção de viabilizar uma interação mais efetiva no contexto de ensino/aprendizagem.
	Pesquisa envolve compreender como e porque um certo tipo de ferramenta será útil	**e envolve também** fazer uma validação de que a ferramenta possui certas propriedades ou certos efeitos, planejando-se cuidadosamente um experimento para medir essas propriedades ou efeitos.
	um que prescreve técnicas de projeto formais,	**e outro que** lida com a seqüência real de estados e ações mentais que ocorrem quando se segue uma técnica de projeto. (...)
	A necessidade de se compartilhar recursos motivou o uso de sistemas distribuídos,	**enquanto** a busca por maior desempenho no processo computacional motivou a utilização do processamento paralelo.
	As técnicas de aferição podem ser: coleta de dados, benchmarks ou construção de protótipos. São mais utilizadas quando o sistema já está implementado, ou em fase final de desenvolvimento.	**Já as técnicas de modelamento podem ser**: modelagem analítica ou simulação.
	Na abordagem retest-all todos os casos de teste utilizados durante a fase de desenvolvimento são empregados.	**Na abordagem** seletiva um subconjunto de casos de teste é selecionado a partir do conjunto original identificando partes do programa modificado que devem ser testadas.
	no caso das regras pontuais, que se utilizam apenas das informações lexicais relativas ao token, está voltada para a identificação dos erros mais comuns cometidos por usuários da língua.	**Nos casos** de concordância nominal e verbal, usa também o recurso das Definite-Clause-Grammars (DCGs), construindo tabelas de ordem de preferência para o uso das regras gramaticais, pois na ausência de informações semânticas sobre o contexto, foram estabelecidos critérios de desambiguação através da freqüência de ocorrência.
	onde a informação evolui naturalmente através de uma série de versões,	**onde** existe a necessidade de acessar versões anteriores da mesma forma que a versão atual
	ou adaptar os processos de desenvolvimento de software existentes	**ou** desenvolver novos processos que suportem a notação proposta pela UML.
	um frame exibe o conteúdo da página original,	**outro frame** mostra o conjunto de versões da página

	Enquanto a multimídia fornece poderosos tipos de dados que facilitam a flexibilidade em expressar a informação,	o hipertexto provê uma estrutura de controle que suporta uma maneira elegante de navegar através dessa estrutura.

Tabela D.27 – Exemplos e marcadores superficiais para a relação SEQUENCE

Ordem	Proposição 1	Proposição 2
NN	Este trabalho propõe um modelo de sumarização automática baseada na utilização da Universal Networking Language, ou UNL.	**A partir** desse modelo, propomos a implementação do protótipo UNLSumm, sigla para UNL Summarizer.
	O caminho alternativo, explorado depois, faz o reconhecimento de padrões a partir do Modelo de Análise do Sistema,	elaborando **a seguir** o projeto avante e implementando o sistema, manualmente, em uma linguagem orientada a objetos.
	A reengenharia com mudança de linguagem é feita tanto de forma automática como de forma manual, obtendo-se sistemas em linguagens orientadas a objetos. (...)	**Após** a engenharia reversa do sistema legado, obtém-se o Modelo de Análise do Sistema, seguindo a orientação a objetos.
	A computação paralela e a computação distribuída surgiram por motivos diferentes. A necessidade de se compartilhar recursos motivou o uso de sistemas distribuídos, enquanto a busca por maior desempenho no processo computacional motivou a utilização do processamento paralelo.	**Atualmente**, as duas áreas têm convergido, de maneira que a combinação entre os dois enfoques computacionais oferece benefícios para ambos os lados. (...)
	Basicamente, são feitas pequenas alterações sintáticas em um programa, gerando um conjunto de programas, denominados mutantes (do programa original P)	**e** constroem-se casos de teste capazes de provocar diferenças de comportamento entre P e seus mutantes.
	O ASiA fornece facilidades gráficas para o usuário desenhar o seu modelo através de redes de filas,	**e a partir** de parametrizações desse modelo, o ambiente gera e executa o programa de simulação.
	os programadores instanciam os elementos da biblioteca em elementos do esquema da aplicação	**e, a seguir**, personalizam esses objetos para adequarem-se às necessidades do aplicativo e da organização.
	com o programa apresentando informações sobre um assunto	**e, em seguida**, fazendo uma série de perguntas a respeito.
	onde o sistema apresenta um tópico sobre determinado assunto (domínio)	**e, logo após**, faz uma série de perguntas a respeito desse assunto.
	Os mutantes são gerados através da aplicação de operadores de mutação, que podem ser estabelecidos de forma a modelar os erros típicos relacionados à técnica de especificação utilizada.	**Em seguida**, a simulação de cada mutante e a comparação dos resultados obtidos com os da especificação em teste contribuem para a análise da adequação do teste e, conseqüentemente, da corretitude da especificação.
	As primeiras implementações, resultantes do surgimento do elemento software, em contrapartida ao elemento hardware, eram realizadas sem qualquer tipo de administração, o que resultava, na maioria das vezes, em prazos esgotados e em custos elevados.	No entanto, **passados alguns anos**, especificamente nas décadas de 1970 e 1980, o software começou a ser desenvolvido para ampla distribuição em um mercado interdisciplinar, época em que vários problemas de funcionamento e eficiência dos produtos começaram a surgir intensamente, em decorrência da falta de

		especificação e planejamento.
	Os testes foram realizados em três implementações do MPI de domínio público que executam sobre a plataforma LINUX. Tal procedimento possibilita uma análise comparativa entre as três implementações, a fim de se determinar, por exemplo, até que ponto uma especificação centrada na eficiência pode garanti-la em qualquer implementação.	**Por fim**, será analisado o comportamento das três implementações face a uma aplicação paralela real, de maneira a comparar-se os resultados obtidos com situações reais de paralelismo.
	Para apoiar a aplicação da Análise de Mutantes neste contexto, Fabbri especificou a ferramenta Proteum-RS (PROduct TEsting Using Mutation for Reactive Systems), e a instanciou para apoiar a validação de especificações baseadas em Máquinas de Estados Finitos, originando a Proteum-RS/FSM.	**Posteriormente**, Sugeta instanciou a Proteum-RS para a validação de Statecharts, originando a Proteum-RS/ST.
	(...) A manutenção envolve qualquer mudança feita Após o software estar em uso.	**Realizada a** manutenção, testes devem ser conduzidos para garantir que a qualidade do software não foi afetada pelas modificações. (...)
	Poderia se imaginar que a utilização do ambiente AMADEUS por um usuário leigo na escrita técnica e língua inglesa começaria pela ferramenta tutorial,	**sendo seguida pela** ferramenta de crítica

Apêndice C – Protocolo de Anotação do RHETALHO

Neste apêndice, apresenta-se o protocolo utilizado para a anotação retórica do RHETALHO, corpus utilizado para a avaliação do DiZer. O protocolo é exibido abaixo, em sua forma original, como foi utilizado pelos anotadores.

Estratégia de Anotação

A anotação retórica deve ser linear, da esquerda para a direita, incremental e modular. Primeiramente, devem-se relacionar todas as orações presentes em uma sentença; depois, todas as sentenças de um parágrafo; por fim, todos os parágrafos do texto devem ser relacionados, formando uma única estrutura retórica. Somente estruturas binárias são permitidas.

Critério de Segmentação

Para a segmentação dos textos, as regras propostas por Carlson and Marcu (2001) devem ser seguidas. Embora essas regras tenham sido definidas para a língua inglesa, elas são genéricas o bastante para serem utilizadas na língua portuguesa.

Se houver discordância entre os anotadores em algum ponto da segmentação, deve-se adotar a segmentação mais genérica e compreensiva.

Determinação de Relações Retóricas (incluindo a determinação de núcleos e satélites)

Deve-se seguir o critério de composicionalidade de Marcu (1997, 2000b). Somente as relações retóricas do conjunto pré-selecionado devem ser consideradas.

Se houver discordância entre os anotadores ao determinar a relação entre pares de segmentos discursivos, uma relação mais genérica deve ser escolhida. Porém, se ambas as relações forem igualmente plausíveis, um terceiro especialista em RST deve ser consultado para decidir a relação mais apropriada.