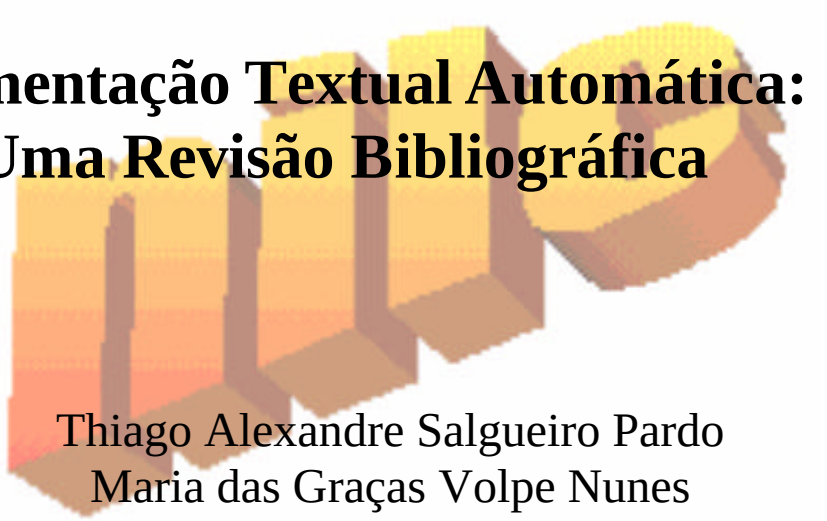
Universidade de São Paulo - USP
Universidade Federal de São Carlos - UFSCar
Universidade Estadual Paulista - UNESP

Segmentação Textual Automática: Uma Revisão Bibliográfica

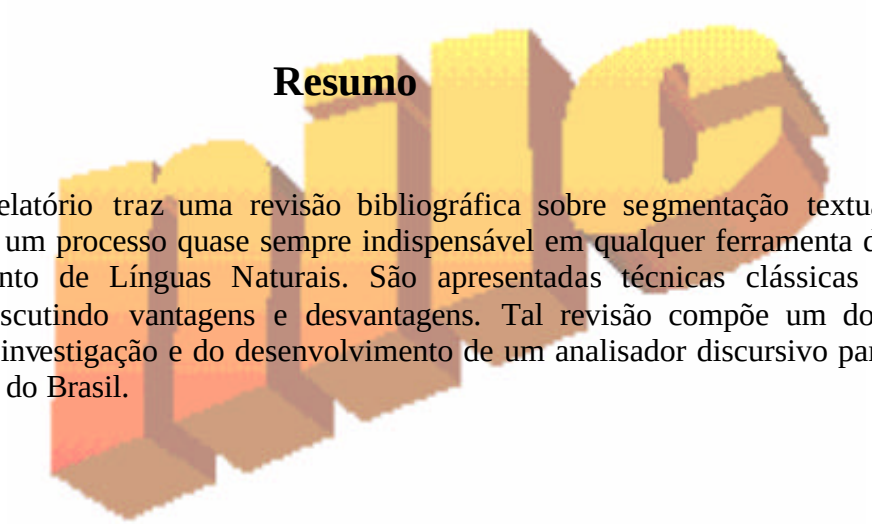


Thiago Alexandre Salgueiro Pardo
Maria das Graças Volpe Nunes

NILC-TR-03-02

Fevereiro, 2003

Série de Relatórios do Núcleo Interinstitucional de Lingüística Computacional
NILC - ICMC-USP, Caixa Postal 668, 13560-970 São Carlos, SP, Brasil



Resumo

Este relatório traz uma revisão bibliográfica sobre segmentação textual automática, um processo quase sempre indispensável em qualquer ferramenta de Processamento de Línguas Naturais. São apresentadas técnicas clássicas e recentes, discutindo vantagens e desvantagens. Tal revisão compõe um dos estágios da investigação e do desenvolvimento de um analisador discursivo para o Português do Brasil.

Este trabalho conta com o apoio financeiro da FAPESP e do CNPq.

ÍNDICE

1. Introdução

A segmentação textual é o processo de segmentar um texto em unidades menores, que podem ser cláusulas, orações[1], sentenças, parágrafos e até mesmo tópicos. A granularidade da segmentação vai depender exclusivamente da aplicação a que se destina a segmentação.

O processo de segmentação é normalmente executado na fase de pré-processamento de ferramentas de Processamento de Línguas Naturais (PLN), já que é uma fase indispensável e essencial para a maioria dessas ferramentas poder realizar suas tarefas específicas, por exemplo, etiquetação morfológica, análise sintática, alinhamento textual, análise retórica, etc.

Apesar de parecer uma tarefa simples, a segmentação se depara com muitos problemas de difícil resolução. Dependendo da granularidade da segmentação, esses problemas podem ter diferentes importâncias. De forma geral, pode-se citar:

1) para qualquer tipo de segmentação, pode haver problemas com o uso do tradicional ponto (“.”): seu uso é ambíguo, pois pode indicar fim de sentença, fim de uma reprodução de uma fala (neste caso, antes de um sinal de aspas), um ponto decimal, uma abreviatura ou até mesmo uma abreviatura no final de uma sentença;
2) para qualquer tipo de segmentação, pode haver problemas com o uso dos pontos de interrogação (“?”) e exclamação (“!”): podem ser usados em fim de sentenças ou dentro de reproduções de fala (novamente, antes do sinal de aspas);
3) para qualquer tipo de segmentação, pode haver problemas com o uso de pontuações como dois pontos (“:”), ponto e vírgula (“;”) e, às vezes, a própria vírgula (“,”): essas pontuações podem ou não indicar novos segmentos;
4) para segmentações clausais e topicais, pode haver problemas com o uso de marcadores discursivos[2]: eles podem ou não indicar a existência de um novo segmento, podendo também ser confundidos com marcadores sentenciais ou pragmáticos;
5) para segmentações topicais, pode haver problemas com referências anafóricas: são difíceis de resolver (computacionalmente).

Com relação ao caso 4, especificamente, marcadores discursivos, como diz sua nomenclatura, são marcadores que possuem alguma função discursiva, indicando relações discursivas entre segmentos. Por exemplo, no trecho de texto abaixo, o marcador “mas” sinaliza uma relação de contraste entre as duas cláusulas:

Ele queria jogar tênis com Janete, *mas* também queria jantar com Suzana.

Entretanto, o marcador nem sempre é discursivo. Por exemplo, o marcador “e”, no trecho abaixo, possui apenas função sentencial, servindo para formar o sujeito composto:

[1] Diferencia-se, neste relatório, cláusula de oração (segundo gramáticas normativas). Cláusula é uma unidade mínima de significado, podendo ter qualquer estruturação interna. Oração, por sua vez, necessariamente deve apresentar um verbo.

[2] No nível do discurso, costuma-se falar sobre proposições em vez de segmentos. Proposições, neste sentido, são unidades conceituais cujas formas superficiais correspondem aos segmentos. Neste trabalho, para uniformidade, o termo “segmento” será usado indiscriminadamente.

Janete *e* Suzana são amigas.

Os marcadores pragmáticos, por sua vez, fazem referência a algum conhecimento que não foi expresso no texto. Por exemplo, no trecho abaixo, o marcador "de novo" faz referência ao conhecimento do leitor do texto, assumindo que este sabe que Janete já jogou tênis anteriormente:

Suzana foi jogar tênis *de novo*.

Há várias abordagens possíveis para se segmentar textos. As mais simples assumem que todo ponto seguido de (pelo menos um) espaço e uma palavra que começa com letra maiúscula indica um novo segmento. Outras, fazendo uso de estudos intensivos de corpora, desenvolvem gramáticas e listas de abreviaturas para que a segmentação seja realizada. Ambos os casos possuem desvantagens: no primeiro caso, não se considera toda a problemática da pontuação (como enumerado anteriormente); no segundo caso, o desenvolvimento da gramática pode ser muito custoso, além de a gramática poder ser ineficiente em tipos e gêneros de textos que não são os dos corpora de treinamento. Além disso, com a grande quantidade de textos on-line, na maioria das vezes, sem nenhum critério de escrita, e o barateamento de mecanismos de escaneamento de textos para o formato digital (com a utilização de OCRs – *Optical Character Recognizers*), pode-se ter textos escritos totalmente em caixa alta ou baixa, caracteres reconhecidos erroneamente e muito ruído (lixo) inserido no texto, dificultando ainda mais o processo de segmentação. Recentemente, grande atenção também tem sido dada à solução do problema da segmentação pelo uso de técnicas de aprendizado de máquina. Essas técnicas, em geral, apresentam um custo relativamente baixo de desenvolvimento, que, neste caso, resume-se ao treinamento e ao teste, já que há muitos sistemas e *toolkits* prontos para tal finalidade.

Este relatório apresenta uma revisão das técnicas de segmentação textual apresentadas na literatura. Esta revisão faz parte da investigação e do desenvolvimento de um analisador discursivo para o Português do Brasil, chamado DiZer-PBr – *Discourse analyZER for BRazilian Portuguese* – que visa automatizar os processos de análise retórica, semântica e intencional de textos científicos. A próxima seção apresenta as principais abordagens propostas na literatura, e as considerações finais são feitas na Seção 3.

2. Revisão bibliográfica

A seguir são descritas diferentes abordagens da literatura para o problema da segmentação textual, a saber: segmentação baseada em regras e heurísticas, em métodos numéricos e estatísticos e em aprendizado de máquina.

2.1. Regras e heurísticas

2.1.1. Expressões regulares

Walker et al. (2001) desenvolveram um sistema de segmentação sentencial baseado no uso de expressões regulares, ou seja, regras que ditam quando segmentar o texto. Afirma-se que as expressões regulares são altamente eficientes nesta tarefa.

Como exemplo, Walker et al. apresentam a expressão regular abaixo, capaz de reconhecer qualquer seqüência de palavras seguida por um ponto, um sinal de pontuação e novamente um ponto:

Sentença → palavra* ponto sinal_de_pontuação ponto

podendo reconhecer sentenças com terminações complexas como .".

He said "I don't like you.".

Outras regras seriam capazes de reconhecer endereços de e-mail, nomes de entidades, etc. A Tabela 1 mostra os resultados obtidos com este método de segmentação, onde:

- *precision* (ou precisão) indica quantos sinais de pontuação foram corretamente identificados em relação a tudo que foi identificado pelo sistema;
- *recall* (ou cobertura) indica quantos sinais de pontuação foram corretamente identificados em relação a tudo que deveria ser identificado pelo sistema;
- *f-measure* é uma medida de eficiência do sistema, combinando *precision* e *recall* em uma medida única.

Tabela 1 – Desempenho do método de Walker et al. com base em expressões regulares

Precision	***Recall***	***F-Measure***
83,43%	95,93%	89,25%

2.1.2. Análise de corpus

Marcu (2000) desenvolveu um processo de segmentação clausal de textos com base nos marcadores discursivos presentes no texto. Ele utilizou os marcadores discursivos por estes estarem diretamente relacionados com a estrutura e o conteúdo textual, isto é, ao mesmo tempo, eles determinam e são determinados pela estrutura e pelo conteúdo textual, servindo para indicar a função discursiva de trechos de textos e suas contribuições para o sucesso da comunicação.

Para realizar a segmentação clausal, Marcu realizou um estudo profundo de corpora de gênero e tipo irrestritos em inglês, compilando uma grande lista de marcadores discursivos, suas posições no texto e quando e como eles causavam a segmentação no texto. Com base nisso, Marcu identificou 11 ações que deveriam ser executadas para segmentar um texto em vista dos marcadores e suas posições no texto. O Quadro 1 mostra alguns exemplos de ações de segmentação associadas à ocorrência dos marcadores discursivos.

Quadro 1 – Marcadores discursivos e ações para segmentação clausal

Marcador	**Posição do marcador**	**Ação**
Although	Começo da sentença	Inserir marca de segmento após a ocorrência da próxima vírgula na sentença
because	Começo da sentença	Inserir marca de segmento imediatamente antes do marcador em questão
for example	Meio da sentença	Não inserir marca alguma
Yet	Começo da sentença	Não inserir marca alguma
where	Começo da sentença	Inserir marca de segmento imediatamente antes do marcador e depois da próxima vírgula na sentença
Sinal de fecha parênteses	Final da sentença	Não inserir marca alguma

Com esse tipo de segmentação, Marcu atingiu o desempenho indicado na Tabela 2.

Tabela 2 – Desempenho do método de Marcu com base em análise de corpus

Precision	***Recall***
90,3%	81,3%

Apesar do bom desempenho do segmentador, é importante ressaltar o grande esforço humano para a compilação dos marcadores discursivos e das ações para segmentação clausal, o que torna esse método proibitivo quando se considera que o processo de segmentação textual normalmente é executado como pré-processamento para as aplicações de PLN.

2.2. Análise numérica e estadística

2.2.1. TextTiling

Palmer e Hearst (1994, 1997) desenvolveram o sistema *TextTiling* para segmentação topical de textos expositivos. Entende-se por segmentação topical a delimitação de um texto nos pontos em que o tema (ou subtópico, como usado pelo próprio autor) do texto muda.

Os autores verificaram que os tópicos de um texto podem ser delimitados de acordo com padrões de co-ocorrência lexical e sua distribuição no texto, ou seja, quando o tópico muda, uma proporção significante de vocabulário muda também. Baseando-se nisso, o sistema *TextTiling* pode utilizar três diferentes técnicas de verificação de co-ocorrência lexical para determinar as delimitações topicais de um texto, a saber: comparação de blocos adjacentes de textos, mudanças de vocabulário e cadeias lexicais.

O *TextTiling* executa os seguintes passos para segmentar um texto:

1) delimitam-se as palavras do texto;

2) aplicam-se às palavras os processos clássicos de *case folding* (deixam-se todas as letras em caixa baixa), de remoção de *stop words* (removem-se palavras muito comuns e insignificantes para a aplicação em questão) e de *stemming* (reduzem-se as palavras às suas formas básicas) propostos por Witten et al. (1994);
3) divide-se o texto em pseudo-sentenças[3] de tamanho fixo;
4) associa-se uma nota para cada pseudo-sentença de acordo com o método de verificação de padrões de co-ocorrência lexical especificado;
5) com base nas notas, as mudanças de tópicos são determinadas e a segmentação é finalizada.

As notas dadas às pseudo-sentenças são calculadas da seguinte forma:

- no caso de comparação de blocos adjacentes de textos, a nota é dada pela soma dos produtos das freqüências com que uma palavra aparece nos blocos sendo comparados, os quais, neste contexto, são formados por conjuntos de pseudo-sentenças. Por exemplo, para os dois primeiros blocos da Figura 1, a nota é calculada como sendo 2*1 (para A) + 1*1 (para B) + 2*1 (para C) + 1*1 (para D) + 1*2 (para E), onde A, B, C, D e E são palavras quaisquer das pseudo-sentenças que formam os blocos;
- no caso de mudanças de vocabulário, as notas são calculadas como a soma das palavras novas introduzidas nos blocos. Tomando o mesmo exemplo anterior (Figura 2), a nota seria calculada como 5 (cinco palavras novas introduzidas: A, B, C, D e E) + 0 (nenhuma palavra nova introduzida);
- no caso de cadeias lexicais, procuram-se seqüências de palavras similares distribuídas entre as pseudo-sentenças e assume-se que a extensão das cadeias de palavras corresponde à extensão do segmento, caso as palavras similares estejam a uma distância máxima limite considerada no *TextTiling*. Para o exemplo anterior (Figura 3), teria-se que todas as instâncias de A encontram-se dentro do limite aceitável, enquanto a terceira instância de B, por exemplo, está muito longe da segunda, contribuindo para determinar, assim, um novo segmento.

[3] Usam-se pseudo-sentenças de tamanho fixo para que seja possível a verificação de padrões de co-ocorrência lexical entre esses trechos de texto. A utilização das próprias sentenças para a verificação causaria problemas, já que as sentenças podem ser muito grandes ou muito pequenas.

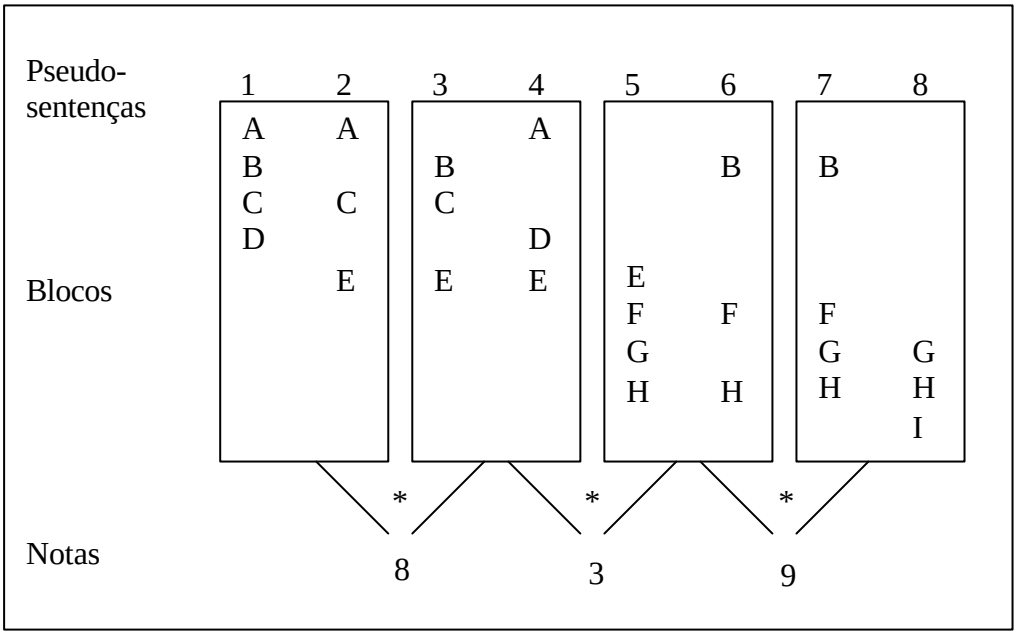


Figura 1 – Comparação de blocos adjacentes

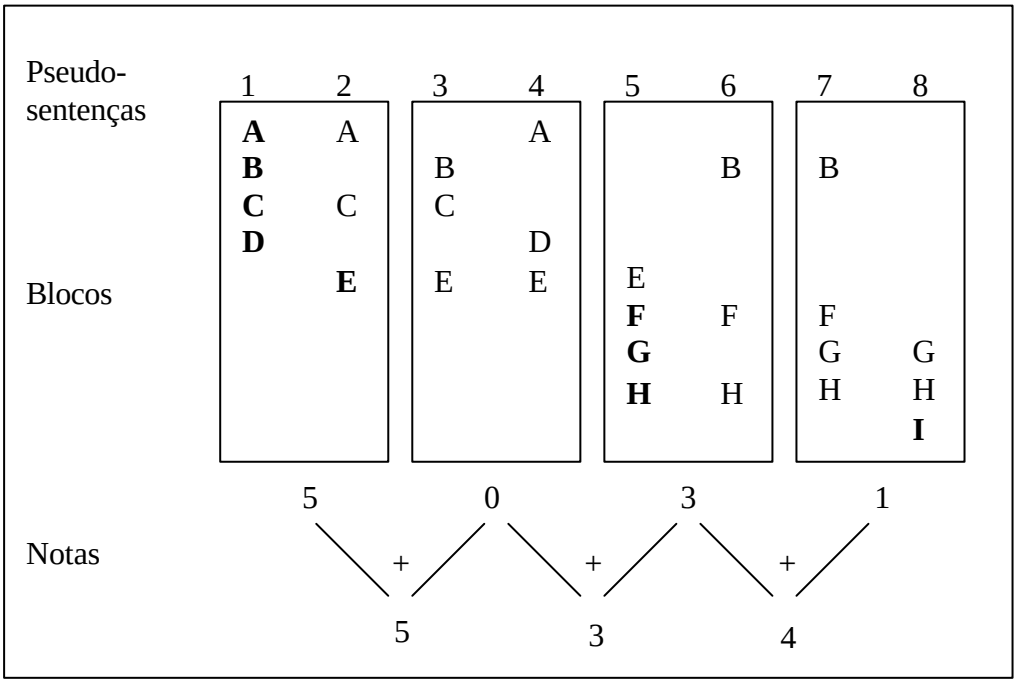


Figura 2 – Mudanças de vocabulário

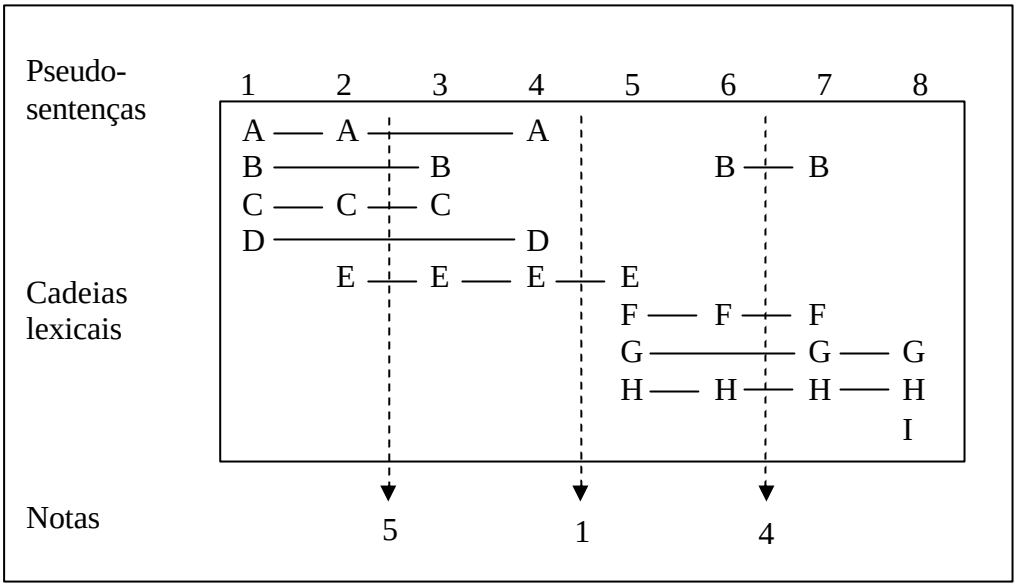


Figura 3 – Cadeias lexicais

Após o cálculo das notas dos blocos (não importando quais dos métodos tenham sido usados), as notas são normalizadas. A seguir, o sistema *TextTiling* plota um gráfico que serve de base para a segmentação do texto. Neste gráfico, a curva representa as notas obtidas para cada par de blocos e os vales da curva (partes da curva com os menores valores no eixo Y) representam a mudança de tópico no texto e, portanto, indicam em que posição o texto deve ser segmentado.

A Tabela 3 apresenta os resultados da avaliação do *TextTilling* feita por seus autores.

Tabela 3 – Desempenho do *TextTilling*

Medida	**Comparação de blocos adjacentes**	**Mudanças de vocabulário**	**Cadeias lexicais**
Precision	71%	58%	64%
Recall	59%	64%	58%

2.2.2. Lexical cohesion profile

Kozima (1993) criou o que chamou de *Lexical Cohesion Profile* (LCP) para realizar a segmentação topical em textos narrativos. Essa segmentação baseia-se puramente em coesão lexical, que, nesse trabalho, é medida pela similaridade das palavras do texto.

O LCP é um registro da coesão lexical de uma seqüência de palavras do texto. De forma semelhante ao que é feito pelo *TextTiling*, os picos e vales em uma seqüência de LCPs podem indicar mudanças de segmentos.

A similaridade entre as palavras de um LCP é calculada com base nos pesos das palavras em uma rede semântica chamada *Paradigme* (Kozima and Furugori, 1993), montada a partir do dicionário de Inglês *Longman Dictionary of Contemporary English*, formando uma estrutura semelhante a um thesaurus. Nessa rede, quanto mais similares duas palavras forem, maior será o peso associado à

combinação delas. Apesar de o autor destacar a eficácia desta técnica, ele não relata qualquer avaliação sistemática.

2.2.3. ROSA

Ferret e Grau (2002) desenvolveram o sistema de segmentação topical ROSA, que, diferentemente dos outros métodos de segmentação, realiza esse processo em duas etapas. A primeira etapa, executada pelo componente do sistema chamado SEGCOHLEX, segmenta o texto de forma similar ao *TextTiling*, utilizando uma rede de colocações para isso. A segunda etapa, executada pelo componente SEGAPSITH, baseada na primeira segmentação, constrói uma representação de tópicos mais aprimorada, agrupando tópicos semelhantes para formar domínios semânticos. A Figura 4 mostra a arquitetura do sistema ROSA.

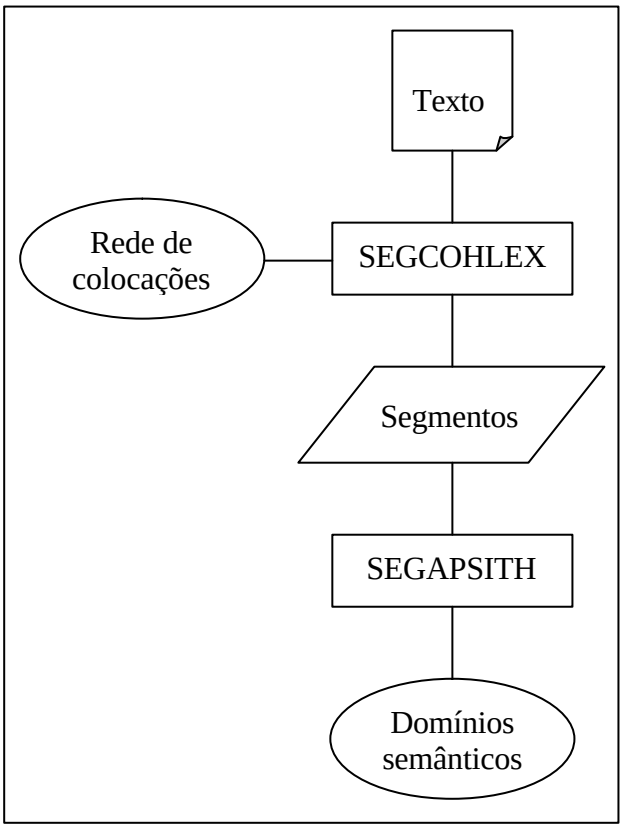


Figura 4 – Arquitetura do sistema ROSA

A rede de colocações é uma rede com 31000 palavras relacionadas entre si. A cada relacionamento na rede está associado um peso correspondente à probabilidade das palavras relacionadas ocorrerem juntas em um texto. De acordo com os autores, essa rede foi construída a partir de um grande corpus de textos jornalísticos.

O componente SEGCOHLEX utiliza a seguinte heurística para segmentar um texto: quanto maior o número de palavras em um trecho de texto relacionado com o mesmo tópico, maior a chance desse trecho de texto constituir um segmento. Isso é feito associando-se às palavras do trecho de texto (1) todas as relações entre as palavras do trecho de texto e pesos possíveis extraídos da rede de colocações e (2) palavras da rede de colocações e seus respectivos pesos, caso exista relação entre

palavras da rede e palavras do trecho de texto. Por fim, com base nas relações e pesos estabelecidos entre as próprias palavras do trecho de texto sob análise e nas relações e pesos estabelecidos entre as palavras do trecho de texto e as palavras da rede de colocação, calcula-se um peso final para o trecho de texto sob análise. Esse mesmo processo é realizado para todos os trechos de texto possíveis. Ao final do processo, plota-se um gráfico com os pesos associados aos trechos de texto, sendo que os vales na curva do gráfico correspondem às mudanças de tópico.

O componente SEGAPSITH recebe como entrada os segmentos gerados pelo componente SEGCOHLEX. Esses segmentos, representantes dos tópicos do texto, são, então, agrupados, formando domínios semânticos. O agrupamento de segmentos para formar domínios se dá pelo cálculo da semelhança dos segmentos em questão. Segmentos muito semelhantes formam um único domínio semântico. Essa semelhança é calculada, basicamente, a partir do número de palavras que co-ocorrem nos segmentos e seus respectivos pesos.

A Tabela 4 mostra o desempenho dos dois componentes de segmentação do sistema ROSA. Como se pode notar, o componente SEGAPSITH se saiu melhor que o SEGCOHLEX, já que se baseia na saída deste, produzindo, assim, uma segmentação mais elaborada.

Tabela 4 – Desempenho dos componentes do ROSA

Componentes	***Recall***	***Precision***	***F-Measure***
SEGCOHLEX	67%	37%	48%
SEGAPSITH	92%	52%	66%

2.2.4. Repetição de palavras e combinação de conhecimentos

Reynar (1994) apresentou um método de segmentação topical baseado puramente na repetição de palavras dentro de documentos. O método se baseia no fato que palavras repetidas ocorrem mais freqüentemente dentro de regiões de um texto que versam sobre um mesmo tópico.

Em seu método, Reynar, inicialmente, processa o texto a segmentar, removendo palavras de classe fechada, removendo palavras de classe aberta muito comuns (*stopwods*) e realizando o processo de *stemming*, ou seja, substituindo as palavras por seus radicais. A seguir, Reynar plota um gráfico representando a distribuição das palavras resultantes do texto pré-processado. Por exemplo, caso uma palavra apareça na posição x e se repita na posição y do texto, os pontos (x,x), (x,y), (y,x), (y,y) serão plotados no gráfico, os quais correspondem ao produto cartesiano das posições da palavra repetida. Ao fazer isso com todas as palavras do texto, o gráfico resultante apresentará intervalos em que nenhum ponto foi plotado, intervalos estes que corresponderão às mudanças de tópico procuradas. A Tabela 5 apresenta os resultados da avaliação deste método.

Tabela 5 – Desempenho do método de repetição de palavras

Precision	***Recall***
54,9%	20,8%

Posteriormente, Reynar (1999) propôs um novo método de segmentação topical baseado em uma combinação de conhecimentos. Para segmentar um texto, Reynar verifica:

- frases indicativas: podem indicar a presença de um novo tópico;
- freqüência de bigramas de palavras: em substituição de palavras-chave isoladas, já que estas podem induzir a erros[4];
- repetição de nomes de entidades: a repetição desses termos pode indicar a continuação do tópico anterior;
- uso de pronomes: a presença de pronomes pode indicar a continuação do tópico anterior, já que os pronomes são elementos altamente coesivos.

Em seu algoritmo, Reynar pesa todas estas características estatisticamente (com base em análise de corpora) para determinar a presença ou não de um novo tópico no texto. Os resultados alcançados são mostrados na Tabela 6.

Tabela 6 – Desempenho do método de combinação de conhecimentos

Precision	***Recall***
59%	60%

2.2.5. Medida de co-seno entre sentenças para cálculo de similaridade

Choi (2000) desenvolveu um método de segmentação topical baseado na medida de co-seno para cálculo de similaridade. Essa medida calcula o grau de similaridade entre duas sentenças com base em suas palavras.

Neste método, o texto é pré-processado da mesma forma como é feito por Reynar (1994). Em seguida, a medida de co-seno para todo par de sentenças do texto é calculada. A medida é calculada pela seguinte fórmula:

$$sim(x,y) = \frac{\sum_j f_{x,j} \times f_{y,j}}{\sqrt{\sum_j f^2{}_{x,j} \times \sum_j f^2{}_{y,j}}}$$

onde x e y representam as sentenças para as quais está se calculando a medida de similaridade e $f_{i,j}$ representa a freqüência da palavra j na sentença i. Com os valores de similaridade calculados, monta-se uma matriz de similaridade, onde a posição (x,y) contém o valor da medida de co-seno para as sentenças x e y do texto. Então, os elementos dessa matriz são ranqueados e normalizados. Por fim, intervalos de baixa similaridade entre as sentenças indicam mudanças de tópico no texto.

Em sua avaliação, Choi relata uma taxa de erro de 13%.

[4] Reynar cita o exemplo da palavra em inglês *plant*: em um trecho do texto, ela poderia compor os termos *wild plant* e *native plant*; em outro trecho, ela poderia compor os termos *chemical plant* e *manufacturing plant*. Apesar de ser a mesma palavra-chave (que, supostamente, indicaria que os trechos de texto em questão são parte de um mesmo tópico), quando composta com outras palavras, indica, na verdade, que o tópico provavelmente mudou.

2.2.6. Teorema de Bayes

Utiyama e Isahara (2001) investigaram o problema da segmentação topical de textos pela aplicação do teorema de Bayes, dado por:

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \times \Pr(A)}{\Pr(B)}$$

No contexto de segmentação, essa fórmula é utilizada da seguinte forma:

$$\Pr(S \mid W) = \frac{\Pr(W \mid S) \times \Pr(S)}{\Pr(W)}$$

onde $W=w_1,w_2...w_n$ é o texto composto por *n* palavras e $S=s_1,s_2...s_m$ são as *m* segmentações presentes no texto. Então, o problema de segmentar o texto consiste em achar a segmentação mais provável do texto, ou seja:

$$Seg = \arg\max \Pr(W \mid S) \times \Pr(S)$$

onde Pr(W|S) é dada pela distribuição das palavras no texto e Pr(S) se baseia no número esperado de segmentos em um texto.

Com esse método, Utiyama e Isahara atingiram uma taxa de erro de 10%.

2.3. Aprendizado de máquina

2.3.1. Árvores de regressão

Riley (1989) desenvolveu um desambiguador sentencial utilizando árvores de regressão[5] como técnica de aprendizado de máquina. Seu sistema foi treinado com os seguintes atributos:

- probabilidade da palavra que precede a pontuação ocorrer no fim de sentenças;
- probabilidade da palavra que segue a pontuação ocorrer no início de sentenças;
- tamanho da palavra que precede a pontuação;
- tamanho da palavra que segue a pontuação;
- caso da palavra que precede a pontuação (maiúscula, minúscula, número);
- caso da palavra que segue a pontuação;
- pontuação após a pontuação (caso exista);
- classe de abreviações de palavras com ponto (“.”).

As duas possíveis classes indicam se a pontuação determina ou não a delimitação de uma sentença. Assim, esse método usa informações sobre as palavras ao redor da pontuação que se quer desambiguar e, portanto, armazena-se em um léxico a probabilidade das palavras ocorrerem ao redor de algum sinal de pontuação. Riley,

[5] Árvores de regressão se caracterizam por utilizarem valores contínuos para os atributos considerados.

em seu trabalho, utilizou um corpus de 25 milhões de palavras para obter as probabilidades e alcançou uma impressionante taxa de erro de 0,2%.

O problema desta técnica reside no tamanho do corpus necessário para se compilar as probabilidades necessárias, tamanho este considerado proibitivo para uma aplicação (i.e., a segmentação) que servirá como pré-processamento para a maioria dos sistemas de PLN.

2.3.2. Satz

Palmer e Hearst (1994, 1997) desenvolveram o sistema de detecção de delimitação sentencial chamado *Satz*[6]. Baseados nos trabalhos de Riley (1989) e Humphrey e Zhou (1989), Palmer e Hearst desenvolveram um novo método de desambiguar a pontuação de textos utilizando algoritmos de aprendizado de máquina. Esse método, em vez de usar as palavras das sentenças para dar o contexto da ocorrência da pontuação, usa as etiquetas morfológicas possíveis das palavras do contexto.

A Figura 5 mostra a arquitetura do sistema Satz.

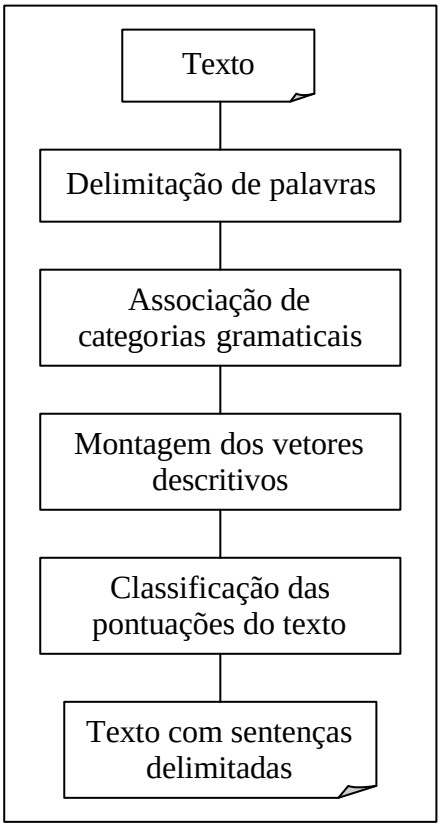


Figura 5 – Arquitetura do *Satz*

De acordo com essa arquitetura, os seguintes passos são realizados para segmentar um texto:

[6] *Satz* significa "sentença" em Alemão.

1) Delimitação das palavras do texto

2) Associação de classes gramaticais: associam-se às palavras do texto a freqüência das palavras ocorrerem como todas as possíveis classes gramaticais consideradas. Por exemplo, a palavra "aprendizado" teria a seguinte distribuição:

substantivo/0.57
verbo/0.36
adjetivo/0.07
advérbio/0
numeral/0
...

As freqüências das palavras são coletadas de um léxico montado a partir de corpora anotados por um *tagger*. No exemplo acima, nos corpora usados para coletar as probabilidades, a palavra "aprendizado" ocorreu como substantivo em 57% dos casos, como verbo em 36% dos casos, como adjetivo em 7% dos casos e não ocorreu como outras classes.
Os autores sugerem que, em casos em que não se tenha a freqüência em que as palavras ocorrem como cada uma das classes gramaticais, pode-se substituir a freqüência das palavras por um indicativo das possíveis classes gramáticas que as palavras podem assumir ("1", se a palavra pode pertencer a uma determinada classe gramatical, e "0", caso contrário). Assim, a distribuição da palavra "aprendizado" teria a seguinte forma:

substantivo/1
verbo/1
adjetivo/1
advérbio/0
numeral/0
...

Em casos de palavras que não constam no léxico ou nos corpora utilizados, estimam-se as possíveis classes gramaticais da palavra por meio de heurísticas simples, por exemplo: (a) se o primeiro caractere da palavra for um número, então a palavra será classificada como numeral; (b) se a palavra começar com uma letra maiúscula, então será classificada como nome próprio; (c) se a palavra contiver um hífen, então será classificada como adjetivo, substantivo e nome próprio.

3) Montagem de vetores descritivos das palavras: para cada palavra do texto, monta-se um vetor descritivo contendo (a) a freqüência da palavra como cada uma das categorias gramaticais consideradas, (b) um indicativo se a palavra começa ou não com letra maiúscula e (c) um indicativo se a palavra segue ou não um sinal de pontuação.

4) Classificação das pontuações do texto: para cada sinal de pontuação encontrado, fornece-se ao classificador (já treinado) os k vetores descritivos que precedem ao

sinal de pontuação em questão e os k vetores descritivos que seguem tal sinal. A saída do classificador indica, então, se este sinal delimita ou não a sentença. Segundo os autores, variando-se o valor de k e realizando-se testes, o melhor valor obtido foi com k=6, isto é, três palavras antes e três palavras depois do sinal de pontuação que se quer desambiguar.

A grande vantagem deste método reside no tamanho do léxico necessário. Diz-se que um léxico com uma média de 5.000 palavras já é suficiente para que o sistema apresente bons resultados na desambiguação da pontuação. Isso é possível pelo fato do algoritmo de aprendizado de máquina não ser treinado com as próprias palavras, mas com suas classes gramaticais associadas, permitindo uma maior generalização do sistema.

O *Satz* utiliza duas técnicas de aprendizado de máquina: redes neurais e árvores de decisão. Para ambos os casos, o treinamento se deu da mesma forma, ou seja, apresentou-se o mesmo conjunto de treinamento com as classes associadas, isto é, se o sinal de pontuação em questão delimitava ou não a sentença, sendo que o conjunto de treinamento consistiu dos k vetores descritivos antes e depois do sinal. Por ser um sistema treinável, a proposta do *Satz* pode ser aplicada para a maioria das línguas naturais.

A Tabela 7 sintetiza os resultados da aplicação do *Satz*, mostrando as taxas de erro do sistema para as línguas inglesa, francesa e alemã, usando tanto redes neurais quanto árvores de decisão. É importante notar a diferença nos tamanhos dos léxicos utilizados e o desempenho do sistema relativo a cada tamanho do léxico.

Tabela 7 – Taxas de erro do *Satz* quando aplicado para várias línguas e corpora

Corpus	**Tamanho do léxico**	**Redes neurais** *Taxa de erro*	**Árvores de decisão** *Taxa de erro*
Wall Street Journal	27,294	1.1%	1.0%
Süddeutsche Zeitung	3,184	1.3%	1.9%
German News	5,037	0.7%	0.7%
Hansards	3,766	0.6%	0.4%

2.3.3. Combinação de conhecimentos

Passonneau e Litman (1993, 1997) e Litman e Passonneau (1995) apresentaram uma nova forma de segmentação prosódica baseada na combinação de conhecimentos de diversas naturezas. Elas desenvolveram essa pesquisa utilizando um corpus transcrito de prosódias. Na grande parte dos casos, a segmentação prosódica corresponde à segmentação clausal, podendo haver segmentos divergentes em casos em que o falante (na prosódia) faz uma pausa em sua fala ou muda sua entonação de voz.

Passonneau e Litman utilizaram árvores de decisão em seu experimento com aprendizado de máquina. Os atributos e seus possíveis valores (indicados entre parênteses) utilizados para treinar o sistema são mostrados abaixo, juntamente com seus significados. As possíveis classes associadas aos atributos foram se existia ou não delimitação prosódica. Como se pode perceber, os atributos são de naturezas diversas.

- Atributos prosódicos:
  - *Before* (*+sentence.final.contour*, *-sentence.final.contour*): entonação do segmento prosódico anterior à possível delimitação (que está sendo classificada pela árvore de decisão);
  - *after* (*+sentence.final.contour*, *-sentence.final.contour*): entonação do segmento prosódico posterior à possível delimitação;
  - *pause* (*true*, *false*): se o segmento prosódico posterior ao possível segmento começa com uma pausa;
  - *duration* (valor numérico contínuo): se *pause* for *true*, então este atributo recebe como valor a duração da pausa.

- Atributos relativos às frases indicativas:
  - cue_1 (*true*, *false*): se o primeiro item lexical do segmento prosódico posterior ao possível segmento começa com alguma palavra indicativa;
  - $word_1$ (*also*, *and*, *anyway*, *basically*, *because*, *but*, *finally*, *first*, *like*, *meanwhile*, *no*, *now*, *oh*, *okay*, *only*, *of*, *see*, *so*, *then*, *well*, *where*, *NA – not applicable*): se cue_1 for *true*, então este atributo recebe como valor o item lexical especificado, caso contrário, recebe NA;
  - cue_2 (*true*, *false*): se cue_1 for *true* e existe uma segunda palavra indicativa após a primeira;
  - $word_2$ (*and*, *anyway*, *because*, *boy*, *but*, *now*, *okay*, *or*, *right*, *so*, *still*, *then*, *NA*): se cue_2 for *true*, então este atributo recebe como valor o item lexical especificado, caso contrário, recebe NA.

- Atributos relativos aos sintagmas nominais:
  - *coref* (*+coref*, *-coref*, *NA*): se o segmento prosódico posterior ao possível segmento contém um sintagma nominal que faz co-referência ao sintagma nominal do segmento prosódico anterior ao possível segmento;
  - *infer* (*+infer*, *-infer*, *NA*): se o segmento prosódico posterior ao possível segmento contém um sintagma nominal que pode ser inferido (por meio de algumas regras de inferências) a partir de um sintagma nominal do segmento prosódico anterior ao possível segmento;
  - *global.pro* (*+global.pro*, *-global.pro*, *NA*): se o segmento prosódico posterior ao possível segmento contém um pronome cujo referente é mencionado em um segmento prosódico anterior à última segmentação detectada.

- Atributo combinado:
  - *cue-prosody* (*complex*, *true*, *false*): se *before=+sentence.final.contour*, *pause=true* e (a) cue_1*=true* e $word_1 \neq$*and* ou (b) cue_1*=true*, $word_1$*=and*, cue_2*=true* e $word_2 \neq$*and*, então *cue-prosody=complex*, caso contrário, *cue-prosody* recebe o mesmo valor de *pause*.

Somente os atributos relativos aos sintagmas nominais foram extraídos manualmente do corpus. Segundo as autoras, os atributos relativos às frases indicativas são facilmente identificados automaticamente, enquanto que os atributos prosódicos são padronizados e seguem uma certa convenção e, portanto, também podem ser identificados automaticamente.

Um sistema foi treinado com os atributos acima isoladamente e em conjunto. Os resultados são mostrados na Tabela 8.

Tabela 8 – Resultados da segmentação com diversos atributos

Atributos	***Recall***	***Precision***	***Error***
Sintagma nominal	66%	25%	17%
Frases indicativas	72%	15%	50%
Pausa	92%	18%	49%
Combinação dos atributos	43%	48%	16%

2.3.4. Aprendizado baseado em instâncias

Stevenson e Gaizauskas (2000) trataram um problema um pouco diferente da segmentação textual da forma como ela é definida aqui. O problema consistia em determinar a pontuação de um texto obtido a partir de um sistema de reconhecimento de fala, o qual produz, normalmente, textos sem pontuação alguma e somente em caixa alta ou baixa[7]. Apesar de ser uma abordagem diferente, ela é revisada aqui por ser uma técnica interessante que também pode ser aplicada para o problema de segmentação enfocado neste relatório.

Stevenson e Gaizauskas utilizaram a técnica de aprendizado de máquina baseada em instâncias, ou, como chamada por eles, algoritmo de aprendizado baseado em memória. Basicamente, são armazenados vários casos de treino e, para descobrir a pontuação de um novo caso, procura-se nesta base de treino o caso mais similar ao novo caso que se quer classificar.

Cada caso consistiu dos seguintes atributos:

- palavra que precede o sinal de pontuação;
- probabilidade da palavra que precede o sinal de pontuação estar no fim de uma sentença;
- classe gramatical[8] da palavra que precede o sinal de pontuação;
- probabilidade da classe gramatical da palavra que precede o sinal de pontuação estar no fim de uma sentença;
- indicação se a palavra que precede o sinal de pontuação é uma *stopword*;
- indicação se a palavra que precede o sinal de pontuação está em caixa alta;
- palavra que segue o sinal de pontuação;
- probabilidade da palavra que segue o sinal de pontuação estar no início de uma sentença;
- classe gramatical da palavra que segue o sinal de pontuação;
- probabilidade da classe gramatical da palavra que segue o sinal de pontuação estar no início de uma sentença;
- indicação se a palavra que segue o sinal de pontuação é uma *stopword*;
- indicação se a palavra que segue o sinal de pontuação está em caixa alta.

[7] Os erros provenientes do mau reconhecimento não foram abordados pelos autores.
[8] Obtida pela aplicação de um *tagger*.

As classes consideradas indicavam simplesmente se os atributos delimitavam ou não uma sentença.

Na avaliação desta técnica, os autores utilizaram dois conjuntos de textos: (1) textos inteiramente em caixa alta obtidos de sistemas de reconhecimento de fala e (2) textos obtidos de sistemas de reconhecimento de fala, mas editados de forma que as palavras ficassem em sua devida caixa. Os resultados para os dois casos são mostrados na Tabela 10.

Tabela 10 – Avaliação da técnica de aprendizado baseada em memória

Caixa	***Precision***	***Recall***	***F-Measure***
Somente alta	78%	75%	76%
Editada	36%	35%	35%

2.3.5. Árvores de decisão

Marcu (2000) utilizou a técnica de árvores de decisão para o processo de segmentação clausal de um texto. O processo funciona da seguinte forma: a cada palavra verificada do texto, utiliza-se a árvore de decisão gerada para indicar se deve ocorrer ou não uma segmentação neste ponto.

Os atributos utilizados por Marcu foram:

- dois lexemas encontrados antes da palavra em foco;
- dois lexemas encontrados depois da palavra em foco;
- as classes gramaticais das duas palavras anteriores à palavra em foco;
- as classes gramaticais das duas palavras posteriores à palavra em foco.

As classes correspondiam às possíveis ações a serem tomadas neste contexto, ou seja, basicamente, segmentar ou não o texto no ponto sendo analisado. Com essa técnica, Marcu atingiu um desempenho médio de 97% para a segmentação clausal.

3. Considerações Finais

Neste relatório, foi realizada uma revisão bibliográfica das técnicas de segmentação textual, a qual consiste em um processo quase sempre indispensável para a maioria das ferramentas de Processamento de Línguas Naturais. Foram apresentadas técnicas antigas e recentes baseadas em regras e heurísticas, em métodos numéricos e estatísticos e em aprendizado de máquina, apontando suas vantagens e desvantagens.

A revisão apresentada faz parte da investigação e do desenvolvimento de um analisador discursivo para o Português do Brasil, chamado DiZer-PBr – *Discourse analyZER for BRazilian Portuguese*[9].

Referências

Choi, F.Y.Y. (2000). Advances in Domain Independent Linear Text Segmentation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, pp. 26-33.

[9] Para mais detalhes sobre este projeto, visite http://www.nilc.icmc.usp.br/~thiago/DiZer.html

Ferret, O. and Grau, B. (2002). A bootstrapping approach for robust topic analysis. In J. Tait, B.K. Boguraev aand C. Jacquemin (eds.), *Natural Language Engineering*. Cambridge. University Press.

Humphrey, T.L. and Zhou, F. (1989). Period Disambiguation Using a Neural Network. In *Proceedings of the International Joint Conference on Neural Networks.* Washington, D.C.

Kozima, H. (1993). Text Segmentation Based on Similarity Between Words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 286-288.

Kozima, H. and Furugori, T. (1993). Similarity Between Words Computed by Spreading Activation on an English Dictionary. In *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 232-239.

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

Litman, D.J. and Passonneau, R.J. (1995). Combining Multiple Knowledge Sources for Discourse Segmentation. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 108-115.

Palmer, D.D. and Hearst, M.A. (1994). Adaptive Sentence Boundary Disambiguation. In *Proceedings of the Conference on Applied Natural Language Processing*. Stuttgart, Germany.

Palmer, D.D. and Hearst, M.A. (1997). Adaptive Multilingual Sentence Boundary Disambiguation. *Computational Linguistics*, Vol. 23, No. 2, pp. 241-267.

Passonneau, R.J. and Litman, D.J. (1993). Intention-Based Segmentation: Human Reliability and Correlation with Linguistic Cues. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 148-155.

Passonneau, R.J. and Litman, D.J. (1997). Discourse Segmentation by Human and Automated Means. In *Computational Linguistics*, Vol. 23, No. 1, pp. 103-139.

Reynar, J.C. (1994). An Automatic Method of Finding Topic Boundaries. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 331-333.

Reynar, J.C. (1999). Statistical Models for Topic Segmentation. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pp. 357-364.

Riley, M.D. (1989). Some Applications of Tree-based Modelling to Speech and Language Indexing. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pp.339-352.

Stevenson, M. and Gaizauskas, R. (2000). Experiments on Sentence Boundary Detection. In *Proceedings of the Applied Natural Language Processing*, pp.84-89. Seattle, WA.

Utiyama, M. and Isahara, H. (2001). A Statistical Model for Domain-Independent Text Segmentation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pp. 491-498.

Walker, D.J.; Clements, D.E.; Darwin, M. and Amtrup, J.W. (2001). Sentence Boundary Detection: A Comparison of Paradigms for Improving MT Quality. In *Proceedings of the 8th Machine Translation Summit*.

Witten, I.H.; Moffat, A.; Bell, T.C. (1994). Managing Gigabytes. *Van Nostrand Reinhold*. New York.