Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

Alinhamento sentencial de textos paralelos: implementação e avaliação de métodos empíricos para o português do Brasil

Helena de Medeiros Caseli
Maria das Graças Volpe Nunes

Nº 176

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Outubro/2002

Índice

Alinhamento Sentencial de Textos Paralelos: Implementação e Avaliação de Métodos Empíricos para o Português do Brasil. [1]

1 Introdução

O Alinhamento de Textos Paralelos é uma subárea de Processamento de Língua Natural (PLN) que tem recebido muita atenção nos últimos anos, devido, principalmente, ao grande número de aplicações para as quais pode ser útil – tradução automática, extração de terminologia, construção de dicionários bilíngües – e de recursos que dela podem ser derivados – léxicos bilíngües, textos paralelos alinhados, dicionários terminológicos.

Basicamente, alinhar dois textos paralelos (textos acompanhados de sua tradução) significa encontrar as correspondências entre eles. Essas correspondências podem se dar em diferentes níveis de resolução: do documento completo até parágrafos, sentenças, palavras ou caracteres. Os dois níveis mais estudados atualmente são o alinhamento de sentenças e o de palavras. Destes, o alinhamento sentencial de textos paralelos foi escolhido para estudo no projeto PESA por possuir maior precisão e apresentar menor complexidade (o que é mais adequado para um primeiro trabalho nessa área).

O projeto PESA[2] (*Portuguese-English Sentence Alignment*) visa estudar, implementar e avaliar diversos métodos de alinhamento sentencial de textos paralelos. Tais métodos estão divididos em três grandes grupos: empíricos, lingüísticos e híbridos.

Os métodos empíricos não utilizam qualquer tipo de informação lingüística em seu processo de alinhamento, apenas informações estatísticas, como a freqüência de ocorrência de palavras e/ou a distribuição delas no texto e técnicas de reconhecimento de padrão que, por exemplo, consideram duas palavras com grafias similares como traduções mútuas. Os métodos lingüísticos, por sua vez, utilizam recursos lingüísticos específicos para as línguas envolvidas, como léxicos, listas de palavras âncoras e glossários. Por fim, os métodos híbridos englobam as duas abordagens anteriores, utilizando os recursos dos métodos empíricos e lingüísticos conjuntamente.

No projeto PESA serão estudados, implementados e avaliados alguns representantes dessas classes de métodos com o objetivo de encontrar pelo menos um que tenha apresentado bons resultados para o par de línguas português brasileiro (PBr) e inglês no domínio específico da computação. Para a etapa de avaliação desses métodos serão submetidos dois *corpora* de textos paralelos compostos por 65 pares de resumos e *abstracts* de trabalhos desenvolvidos no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, em São Carlos. Esses *corpora* receberam a denominação de *corpus* autêntico e *corpus* pré-editado, isso porque no

[1] Este trabalho foi apoiado por CNPq e CAPES.
[2] Mais detalhes em Caseli (2002) disponível em http://www.nilc.icmc.usp.br/nilc/projects/pesa.htm

primeiro os textos estão na forma como foram originalmente redigidos e, no segundo, os textos foram processados por um tradutor humano para a remoção de erros do PBr e da tradução para o inglês.

Entre todos os métodos que fazem parte do projeto PESA, este relatório traz os detalhes da implementação e avaliação dos dois representantes da classe de métodos empíricos: o GC (Gale e Church, 1991; Gale e Church, 1993) e o GMA (Melamed, 2000).

O método GC baseia-se em um modelo estatístico simples que leva em consideração apenas o tamanho das sentenças, em caracteres, para determinar as correspondências entre elas. O método GMA, por sua vez, utiliza a técnica de reconhecimento de padrão para determinar os pares de palavras com grafias similares e os considera pontos de correspondência das sentenças a serem alinhadas.

A próxima seção (Seção 2) especifica o formato de entrada e saída dos arquivos a serem alinhados pelos métodos de alinhamento sentencial segundo as especificações do projeto PESA. A Seção 3 traz uma descrição resumida do método GC, seu processo de alinhamento e alguns detalhes de sua implementação (Subseção 3.1). De forma análoga, a Seção 4 apresenta o método GMA, seu processo de alinhamento (Subseção 4.1) e explica, na subseção 4.2, o processo de otimização de seus parâmetros para adequação deste método ás línguas envolvidas.

Os resultados da avaliação destes métodos utilizando os *corpora* autêntico e pré-editado são apresentados na Seção 5. A última seção (Seção 6) traz uma breve conclusão sobre este trabalho e as sugestões de melhorias para os métodos em um futuro próximo.

2 Entrada e Saída

Primeiramente, os textos a serem alinhados devem, necessariamente, ser a tradução um do outro, ou seja, textos paralelos. O texto original e sua tradução recebem as denominações de texto fonte e texto alvo, respectivamente. Assim, a entrada para todo método de alinhamento de textos paralelos é formada por um texto fonte e seu correspondente texto alvo. Para facilitar o processo de alinhamento e permitir que todos os textos paralelos de um *corpus* paralelo sejam alinhados com uma só chamada ao método, um arquivo contendo os caminhos para esses pares de textos foi criado. Cada linha desse arquivo contém os caminhos para o texto fonte e o texto alvo, separados por um espaço em branco, como mostrado a seguir:

C:\Temp\Corpus\Resumos\art1R.txt C:\Temp\Corpus\Abstracts\art1A.txt
C:\Temp\Corpus\Resumos\art2R.txt C:\Temp\Corpus\Abstracts\art2A.txt
…

Esse arquivo pode ser gerado utilizando o módulo de geração de *corpus* paralelo da ferramenta de pré-processamento de textos TagAlign[3] e será referenciado no restante desse texto como <corpus paralelo>.

Além do <corpus paralelo> outros três parâmetros podem ser passados na chamada dos métodos analisados neste relatório: as etiquetas de identificação do texto e de marcação de fronteiras de parágrafos e sentenças. Se essas três etiquetas não forem explicitamente informadas, o método considerará as etiquetas padrão: text, p e s, respectivamente.

Cada um dos textos indicados no <corpus paralelo> deve estar marcado com essas etiquetas, o que pode ser feito com o auxílio da TagAlign. A etiqueta <text> indica o início do texto e possui dois atributos que identificam a língua na qual o texto foi escrito (lang) e o nome do arquivo no qual este texto está armazenado (id). A Figura 1 traz um exemplo de um texto fonte etiquetado.

```
<text lang=pt id=art1R>
<p><s>Neste artigo é apresentada uma ferramenta para validação e verificação de
requisitos.</s><s>Essa ferramenta suporta a abordagem ERACE.</s><s>Tal abordagem parte do
documento de requisitos do sistema e propõem  a especificação das interações entre o sistema e seus
agentes (cenários), e então os cenários são especificados detalhadamente.</s><s>Também são
apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise,
exemplificadas através do estudo de caso apresentado.</s>
</p>
</text>
```

Figura 1 – Exemplo de um texto fonte etiquetado.

Os textos de entrada, após serem alinhados, serão salvos em arquivos com o mesmo nome do original, porém com a extensão **.al**. Nesses arquivos, as correspondências entre as sentenças

[3] Mais detalhes em (Caseli, Feltrim e Nunes, 2002) disponível em http://www.nilc.icmc.usp.br/nilc/publications.htm.

serão indicadas por dois atributos – id e corresp – inseridos na etiqueta inicial das sentenças. O atributo id contém um identificador único para a sentença e o corresp, os identificadores das sentenças correspondentes a ela. A Figura 2 mostra um exemplo de um par de textos paralelos antes e depois do alinhamento com o método GC.

<text lang=pt id=art1R> <s>Neste artigo é apresentada uma ferramenta para validação e verificação de requisitos.</s><s>Essa ferramenta suporta a abordagem ERACE.</s><s>Tal abordagem parte do documento de requisitos do sistema e propõem a especificação das interações entre o sistema e seus agentes (cenários), e então os cenários são especificados detalhadamente.</s><s>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s> </text>	<text lang=pt id=art1R> <s id=art1R.1.s1 corresp=art1A.1.s1>Neste artigo é apresentada uma ferramenta para validação e verificação de requisitos.</s><s id=art1R.1.s2 corresp=art1A.1.s2>Essa ferramenta suporta a abordagem ERACE.</s><s id=art1R.1.s3 corresp=art1A.1.s3>Tal abordagem parte do documento de requisitos do sistema e propõem a especificação das interações entre o sistema e seus agentes (cenários), e então os cenários são especificados detalhadamente.</s><s id=art1R.1.s4 corresp=art1A.1.s4>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s> </text>
<text lang=en id=art1A> <s>A tool to support requirements trading is presented.</s><s>The tool supports the ERACE approach.</s><s>This approach starts from the system's requirement document and proposes to specify interactions between the system and its agents (scenarios), and then the scenarios are specified in detail.</s><s>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s>An example to illustrates the approach is also presented.</s> </text>	<text lang=en id=art1A> <s id=art1A.1.s1 corresp=art1R.1.s1>A tool to support requirements trading is presented.</s><s id=art1A.1.s2 corresp=art1R.1.s2>The tool supports the ERACE approach.</s><s id=art1A.1.s3 corresp=art1R.1.s3>This approach starts from the system's requirement document and proposes to specify interactions between the system and its agents (scenarios), and then the scenarios are specified in detail.</s><s id=art1A.1.s4 corresp=art1R.1.s4>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s id=art1A.1.s5 corresp="">An example to illustrates the approach is also presented.</s> </text>

Figura 2 – Exemplo de um par de textos paralelos antes e depois do alinhamento.

Além dos textos de entrada alinhados, outros arquivos de saída podem ser gerados de acordo com o método utilizado. As seções a seguir trazem mais detalhes a respeito dos métodos analisados neste trabalho.

3 O Método GC

O método de Gale e Church (1991, 1993) foi um dos primeiros métodos de alinhamento sentencial propostos à comunidade científica, juntamente com o método de Kay e Röscheisen (1988, 1993). Os dois são representantes da classe de métodos empíricos e, até hoje, servem de base para diversos novos métodos nessa área.

Com já dito, o método GC baseia-se em um modelo estatístico simples que leva em consideração apenas o tamanho das sentenças, em caracteres, para determinar as correspondências entre elas. Esse método parte do pressuposto de que o tamanho de uma sentença no texto fonte está fortemente relacionado ao tamanho de sua tradução no texto alvo: sentenças curtas tendem a ter traduções curtas, e sentenças longas, traduções longas. Além disso, verifica-se uma taxa relativamente fixa entre os tamanhos das sentenças em quaisquer duas línguas, medida em número de caracteres ou de palavras. No caso do par de línguas estudado neste trabalho constatou-se que os textos dos *corpora* utilizados na avaliação possuem uma taxa média de 0,89 caractere em inglês para cada caractere em PBr.

Em uma primeira avaliação realizada com um *corpus* composto por relatórios econômicos do *Union Bank of Switzerland* (UBS) em três idiomas: Inglês, Francês e Alemão, o método GC alinhou corretamente todas as sentenças, com exceção de 4% delas. Além disso, desse *corpus* foi extraído um sub-corpus composto por 80% dos textos que obtiveram melhor precisão no alinhamento e a taxa de erro para esse sub-corpus baixou de 4% para 0,7%. Os detalhes de como essa avaliação se realizou não são fornecidos.

Em outras ocasiões, como na avaliação de diversos métodos de alinhamento de textos paralelos nos níveis sentencial e lexical realizada em (Véronis e Langlais, 2000), o método GC foi utilizado como base para comparação dos métodos sentencial apresentando valores entre 62 e 82% para *f-measure* (explicada na seção 5), de acordo com os gêneros dos textos envolvidos. Por esses e outros motivos, o método GC foi escolhido para integrar o projeto PESA como um dos representantes dos métodos empíricos.

Neste trabalho, o método GC foi implementado em Perl com algumas pequenas alterações em relação ao código original apresentado em (Gale e Church, 1993). A subseção seguinte (3.1) traz uma explicação geral do processo de alinhamento e alguns detalhes de sua implementação e adequação aos requisitos do projeto PESA.

3.1 O Alinhamento

Resumidamente, o processo de alinhamento do método GC possui dois passos. Primeiro, os parágrafos dos bitextos são alinhados entre si e, então, as sentenças dentro dos parágrafos são

alinhadas. O alinhamento de parágrafos pode ser automático, acompanhado de uma verificação manual. Porém, o método tem a limitação de só alinhar textos com o mesmo número de parágrafos.

O alinhamento de sentenças utiliza a técnica de programação dinâmica freqüentemente empregada para alinhar duas seqüências de símbolos, como no caso de códigos genéticos de duas espécies diferentes. No alinhamento sentencial, a programação dinâmica mede a similaridade de duas sentenças verificando a facilidade de transformar uma na outra. Nesse processo, são aplicadas seis operações básicas: remoção, inserção, substituição, contração, expansão e união.

Assim, dado um número de alinhamentos possíveis, o método GC busca o "melhor" alinhamento que englobe o maior número de sentenças em uma determinada vizinhança. O melhor alinhamento é determinado utilizando-se: uma medida de distância para comparar dois elementos individuais dentro das seqüências, e um algoritmo de programação dinâmica para minimizar as distâncias totais entre os elementos alinhados dentro de duas seqüências. Em outras palavras, o processo de alinhamento tenta encontrar as sentenças mais "próximas", ou seja, as sentenças candidatas à tradução.

A chamada inicial ao método GC tem o seguinte formato:

align <corpus paralelo> [<etiquetas>]

na qual, <corpus paralelo> é um arquivo com os caminhos para os arquivos com os textos fonte e alvo (vide Seção 2).

O primeiro passo do método é alinhar os parágrafos. No caso dos *corpora* usados na avaliação descrita na Seção 5, essa tarefa foi muito simples uma vez que a maioria dos pares de resumos e *abstracts* possui apenas um parágrafo. Além disso, devido à limitação do método de só alinhar textos com o mesmo número de parágrafos os que possuíam número diferente não foram alinhados e por isso foram excluídos da avaliação reduzindo-se o número de bitextos disponíveis nessa tarefa de 65 para 64 no *corpus* autêntico e para 63 no pré-editado.

O objetivo do alinhamento sentencial é encontrar o alinhamento com maior probabilidade dado um conjunto de possibilidades. Para isso, calcula-se uma medida de distância que verifica a probabilidade de uma sentença na língua L1 ser a tradução de um conjunto (possivelmente vazio ou unitário) de sentenças na língua L2. Essa probabilidade é calculada baseada em dois parâmetros: a média e a variância do número de caracteres na língua L2 por caractere na língua L1.

A média (c) pode ser estimada pela soma do número de caracteres no texto escrito em L2 dividida pelo número de caracteres no texto escrito em L1:

$$c = \frac{NúmeroDeCaracteresEmL2}{NúmeroDeCaracteresEmL1} \quad (1)$$

A variância (s^2) é estimada em função dos comprimentos dos textos sendo alinhados e é

determinada pela inclinação da linha de regressão robusta considerando a relação entre os tamanhos dos parágrafos em inglês (valores do eixo x) e o quadrado da diferença de tamanho entre os parágrafos nas duas línguas (valores do eixo y).

Dessa forma, a medida de distância (*d*) baseia-se em um modelo probabilístico o que, segundo os autores, permite que a informação seja combinada de uma forma consistente:

$$d = -\log Prob(match|\delta) \quad (2)$$

onde δ depende dos tamanhos das duas porções dos textos sob consideração e o log é utilizado apenas para garantir que as distâncias produzirão resultados desejados (entre 0 e 1).

Essa medida de distância parte do pressuposto de que cada caractere em uma língua L1 dá origem a um número randômico de caracteres em outra língua, L2. Assume-se que estas variáveis randômicas são independentes e identicamente distribuídas com uma distribuição normal. δ é então:

$$\delta = \frac{(l_2 - l_1 c)}{\sqrt{l_1 s^2}} \quad (3)$$

onde l_2 e l_1 são os tamanhos (em caracteres) das porções sob consideração nos textos alvo e fonte, respectivamente; c é o número de caracteres esperados em L2 por caractere em L1 e s^2 é a variância do número de caracteres em L2 por caractere em L1.

Todos os cálculos envolvidos na determinação da distância *d* são efetuados na sub-rotina *two_side_distance*. Nela são utilizadas três constantes referentes às penalidades para os alinhamentos diferentes de 1:1, ou seja, 0:1 ou 1:0, 1:2 ou 2:1 e 2:2. Essas constantes foram calculadas com base na probabilidade de ocorrência de cada tipo de alinhamento nos dois *corpora* disponíveis para a avaliação: o *corpus* autêntico e o *corpus* pré-editado resultando nos valores apresentados na Tabela 1.

Tabela 1. Probabilidades dos alinhamentos.

	***Corpus* Autêntico**		***Corpus* Pré-editado**	
Categoria	**Freqüência**	**Probabilidade**	**Freqüência**	**Probabilidade**
1:1	352	0,873	394	0,949
1:0 ou 0:1	6	0,015	2	0,005
2:1 ou 1:2	41	0,102	17	0,041
2:2	4	0,0099	2	0,005

Uma outra categoria não incluída na Tabela 1 e presente no *corpus* autêntico é a 2:3. Esta foi excluída da análise de probabilidade por não ser tratada pelo método GC que, por razões computacionais, possui a limitação de alinhar apenas pares *m:n* com $0 \leq m, n \leq 2$, possibilitando,

assim, que o alinhamento ótimo seja eficientemente computado aplicando-se o algoritmo de programação dinâmica convencional. O único alinhamento do tipo 2:3 encontrado foi desconsiderado nessa análise.

A sub-rotina *two_side_distance* recebe quatro argumentos – x_1, y_1, x_2, y_2 –correspondentes às sentenças nos textos fonte e alvo, e calcula a distância de acordo com esses valores sendo:

1. $d(x_1, y_1; 0, 0)$ o custo da substituição de x_1 por y_1,
2. $d(x_1, 0; 0, 0)$ o custo da remoção de x_1,
3. $d(0, y_1; 0, 0)$ o custo da inserção de y_1,
4. $d(x_1, y_1; x_2, 0)$ o custo da contração de x_1 e x_2 para y_1,
5. $d(x_1, y_1; 0, y_2)$ o custo da expansão de x_1 para y_1 e y_2, e
6. $d(x_1, y_1; x_2, y_2)$ o custo da união de x_1 e x_2 correspondendo a y_1 e y_2.

Dessa forma, a probabilidade para cada par de sentenças proposto é calculada em relação aos comprimentos das sentenças dos dois textos e da variância dessa relação e esses valores são submetidos a um algoritmo de programação dinâmica.

A programação dinâmica é uma técnica para otimização de problemas nos quais a solução final é construída a partir de sucessivas escolhas locais, mas não significando que uma escolha local ótima fará parte da solução final ótima (Campbell, Chatterjee e Dawkins, 1998).

No alinhamento de textos paralelos, o algoritmo de programação dinâmica tenta encontrar o alinhamento com a menor distância dentro da maior região (ou vizinhança). No método GC são calculadas seis distâncias (descritas acima) para as sentenças fonte, $s_1 \ldots s_i$, e suas traduções, $t_1 \ldots t_j$, e considera-se a melhor solução, D(i,j), a distância mínima entre todas as combinações possíveis. Assim:

$$D(i,j) = \min \begin{cases} D(i, j-1) + d(0, t_j; 0, 0) \\ D(i-1, j) + d(s_i, 0; 0, 0) \\ D(i-1, j-1) + d(s_i, t_j; 0, 0) \\ D(i-1, j-2) + d(s_i, t_j; 0, t_j - 1) \\ D(i-2, j-1) + d(s_i, t_j; s_i - 1, 0) \\ D(i-2, j-2) + d(s_i, t_j; s_i - 1, t_j - 1) \end{cases}$$

Por fim, o par que apresentar a maior probabilidade é escolhido. O caso mais comum acontece quando uma sentença no texto fonte corresponde a exatamente uma sentença no texto alvo, 1:1. Mas casos de omissão, 1:0, adição, 0:1, ou fusão de complexidade variável, m:n com $0 \leq m, n \leq 2$, também são encontrados.

Os resultados da avaliação deste método são apresentados na Seção 5.

4 O Método GMA

O *Geometric Mapping and Alignment* (GMA) é um método empírico de alinhamento sentencial de textos paralelos que utiliza dois algoritmos em seu processo de alinhamento: o *Smooth Injective Map Recognizer* (SIMR) e o *Geometric Segment Alignment* (GSA) (Melamed, 2000). Embora pertença à mesma classe que o método GC – a classe dos métodos empíricos – o GMA possui critérios de alinhamento diferentes deste. Enquanto o primeiro baseia-se na idéia de correlação entre os tamanhos das sentenças a serem alinhadas, o segundo utiliza a técnica de reconhecimento de padrão para mapear os pontos de correspondência entre os dois textos (SIMR) e, a partir dos mapeamentos resultantes, alinhar as sentenças (GSA).

O GMA é um método de código aberto e a versão usada neste trabalho é a 1.1.2. O código foi escrito em Perl e C++ e, além dos originais, outros programas foram escritos em Perl para atender as necessidades do projeto PESA.

Em uma avaliação realizada em (Melamed, 2000) utilizando o *corpus* Hansard com textos paralelos Inglês-Francês, o método GMA apontou uma precisão de 97,7% a 98,5%, de acordo com o tipo de texto analisado e a existência de um alinhamento prévio no nível de parágrafos.

Em uma outra avaliação apresentada pelo projeto ARCADE, no qual o GMA foi utilizado, a precisão obtida foi de 94,2% em textos técnicos sem correspondências cruzadas, ou seja, as correspondências entre sentenças obedecem se dão na mesma ordem em que elas aparecem nos textos. Além disso, sua performance foi a melhor entre os participantes que não confiavam em recursos externos como léxicos ou glossários.

A próxima Subseção (4.1) apresenta o processo de alinhamento do método GMA e as alterações feitas no método original para adaptá-lo aos requisitos do projeto PESA.

4.1 O Alinhamento

Como mencionado na subseção anterior, o GMA utiliza dois algoritmos para alinhar os textos paralelos: o SIMR e o GSA.

O SIMR é um algoritmo genérico de reconhecimento de padrão particularmente bem-sucedido para mapeamento de correspondências em bitextos. O objetivo desse algoritmo é identificar palavras no texto alvo similares a palavras no texto fonte (segundo critérios explicados mais tarde) e retornar pares de coordenadas (*x*,*y*) indicando que na posição *y* do texto alvo existe uma palavra que pode ser considerada a tradução de uma palavra na posição *x* do texto fonte.

O mapeamento resultante, também denominado mapeamento de bitexto, está longe de ser considerado um bom alinhamento de palavras, mas é muito eficiente quando usado como passo intermediário no processo de alinhamento de segmentos.

Geometricamente, o problema de mapeamento de bitexto utilizando reconhecimento de padrão pode ser compreendido através da ilustração na Figura 3.

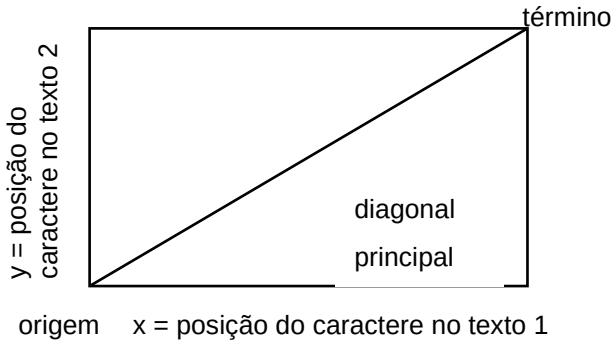

Figura 3 – Os bitextos são dispostos em dois eixos perpendiculares formando um retângulo que recebe o nome de *espaço do bitexto*. Por convenção, a cada palavra presente nos textos é atribuída a posição de seu caractere mediano (Melamed, 2000).

A determinação dos pontos (*x*,*y*) que formam o mapeamento do bitexto engloba duas fases: a fase de geração e a fase de reconhecimento. Na fase de geração, podem-se utilizar diversos recursos para determinar se duas palavras são a tradução uma da outra como uma lista de palavras âncoras (lista multilingüe de palavras que são traduções mútuas) ou uma medida baseada em cognatos que, embora seja mais simples, apresenta resultados satisfatórios no caso de línguas similares.

Na avaliação do método GMA descrita aqui foi utilizada uma medida baseada em cognatos denominada *Longest Common Subsequence Ratio* (ou LCSR). A LCSR de duas palavras é a razão do tamanho da maior subseqüência comum (não necessariamente contínua), denotada por LCS (*Longest Common Subsequence*), e o tamanho da maior palavra. Se o valor de LCSR para duas palavras, A e B, for maior (ou igual) do que um determinado valor limite, então A e B são consideradas cognatas. Assim, o LCSR(A,B) é calculado como segue:

$$LCSR(A,B) = \frac{tamanho[LCS(A,B)]}{\max[tamanho(A), tamanho(B)]} \quad (4)$$

Por exemplo, a palavra *medicina* possui 7 caracteres que aparecem na mesma ordem na palavra *medicine*. Assim, o LCSR para estas palavras é 7/8 (ou 0,875). Já o LCSR de *mensagem* e *message* é apenas 4/8 (ou 0,5). O valor limite para o LCSR é um dos parâmetros do SIMR e foi otimizado junto com os demais como explicado na subseção 4.2.

Nesse processo de determinação dos pontos de correspondência, pode-se utilizar uma *stoplist* para cada uma das línguas envolvidas, ou seja, uma lista composta por palavras muito freqüentes. Essas palavras, por exemplo o artigo “a”, geralmente geram mais de um ponto de correspondência na mesma coluna ou linha do espaço do bitexto. Uma vez que apenas um ponto de correspondência em cada linha e coluna pode estar correto, todos os outros são ruídos. Para evitar a geração de ruído foram utilizadas *stoplists* para o inglês (fornecida com o método) e para o PBr

(formada por artigos, preposições, pronomes e alguns advérbios).

Após a geração dos pontos de acordo com a LCSR, uma filtragem é realizada para a remoção daqueles que podem gerar ruídos. O filtro aplicado neste caso baseia-se no parâmetro de *nível de ambigüidade máximo do ponto*. Para cada ponto *p=(x,y)* calcula-se o nível de ambigüidade (NA) da seguinte forma:

$$NA(p) = X + Y - 2 \quad (5)$$

em que X e Y são o número de pontos na coluna *x* e o número de pontos na linha *y*, respectivamente. O valor calculado é comparado com o parâmetro e os pontos que ultrapassarem este valor são ignorados automaticamente.

Depois da filtragem de ruído, tem-se a fase de reconhecimento na qual cadeias (seqüências lineares) de pontos são testadas em relação a três propriedades: injectividade[4], linearidade e inclinação constante.

A propriedade de injectividade garante que não existem dois pontos em uma cadeia com as mesmas coordenadas *x* ou *y*. As cadeias que não obedecem esta propriedade são automaticamente rejeitadas. A propriedade de linearidade é entendida como a tendência dos pontos a se alinhar e é verificada calculando-se a raiz da distância média ao quadrado dos pontos da cadeia a partir da linha de mínimos quadrados dessa cadeia. Se a distância exceder o parâmetro de *dispersão máxima da cadeia ponto*, a cadeia é rejeitada.

Por fim, verifica-se a propriedade de inclinação constante: quando a inclinação de uma cadeia se aproxima da inclinação do bitexto, ou seja, de sua diagonal principal. Para isso compara-se o ângulo da linha de mínimos quadrados de cada cadeia à arctangente da inclinação do bitexto. Se a diferença exceder o parâmetro de *desvio máximo do ângulo* a cadeia é rejeitada.

Além dos parâmetros citados existe outro de fundamental importância em todo o processo de determinação das cadeias de pontos de correspondência: o tamanho da cadeia. O SIMR especifica um tamanho fixo (k) para a cadeia, com $6 \leq k \leq 11$, sendo que o valor exato de k depende da língua e deve ser otimizado junto com os outros parâmetros como mostrado na subseção 4.2. Todos esses parâmetros diminuem o espaço de busca por cadeias mantendo a complexidade do algoritmo dentro dos padrões aceitáveis.

Assim, o resultado do SIMR é um mapeamento dos pontos de correspondência dos textos fonte e alvo como mostrado na Figura 4. Esse mapeamento, juntamente com informações sobre as fronteiras dos segmentos

[4] Tradução encontrada na área, mas não abonada, de injectivity.

O outro algoritmo utilizado pelo GMA é o GSA: um algoritmo que alinha segmentos de qualquer tamanho a partir dos mapeamentos retornados pelo SIMR e informações sobre as fronteiras dos segmentos (sentenças, no caso do alinhamento sentencial). Essas informações são a entrada do GSA e podem ser geometricamente visualizadas na Figura 4, onde cada célula representa o produto de duas sentenças, uma de cada texto. Um ponto de correspondência dentro da célula (*X*, *y*) indica que alguma palavra na sentença *X* corresponde a alguma palavra na sentença *y*, isto é, as sentenças *X* e *y* correspondem. Dessa forma, se as sentenças (X_1, ... X_n) alinham com as sentenças (y_1, ... y_m), então [(X_1, ... X_n), (y_1, ... y_m)] constitui um *bloco alinhado*. Os blocos alinhados na Figura 4 são demarcados com linhas sólidas.

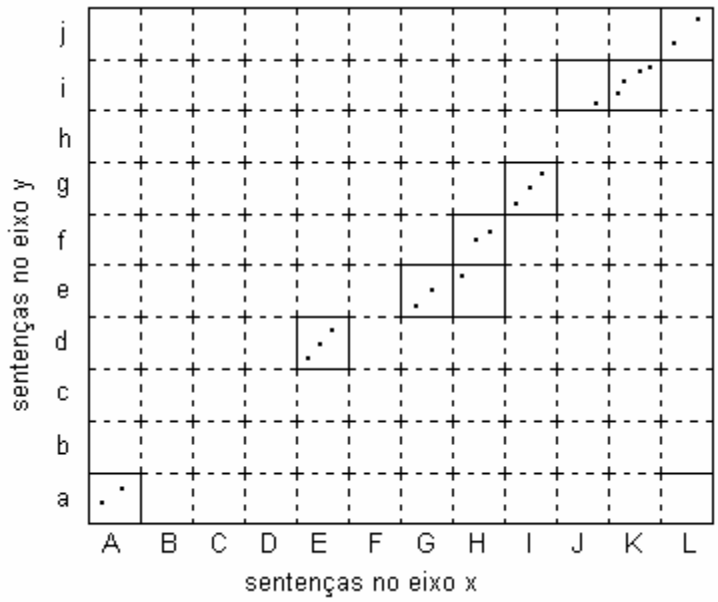

Figura 4 – Informações de entrada para o algoritmo GSA (Melamed, 2000).

O primeiro passo do GSA é arranjar todas as células que contém pontos de correspondência em retângulos que não se sobreponham. Para isso as seguintes operações são executadas. Primeiro, se a entrada contém, por exemplo, os pares (*G*,*e*), (*H*,*e*), e (*H*,*f*) como ilustrado na Figura 4, então o GSA adiciona o par (*G*,*f*). Depois, o GSA força todos os segmentos a serem contínuos: se a sentença *Y* corresponde às sentenças *x* e *z*, mas não à *y*, sendo que a ordem delas no texto é *x*, *y* e *z*, então o par (*Y*,*y*) é adicionado.

Os passos seguintes tentam reduzir os erros produzidos pela ausência de pontos de correspondência em segmentos de um dos textos (por exemplo, as sentenças (B,C,D) e (b,c) na Figura 4), pela existência de alinhamentos do tipo 1:n, onde n > 1 (por exemplo, as células (H,e) e (H,f) na Figura 4), ou pela combinação desses casos. Para isso o GSA faz o realinhamento usando um método baseado em tamanho (o GC, por exemplo). Se esse realinhamento ultrapassar um nível de confiança pré-estabelecido, o GSA aceita o resultado produzido pelo realinhamento, caso contrário, o alinhamento indicado pelos pontos de correspondência do SIMR é mantido.

Além dos arquivos fonte originais da versão 1.2 do GMA, novos programas foram

implementados para acrescentar as especificações exigidas pelo PESA. Um novo programa foi escrito para ser o "programa principal" e referenciar os outros presentes no código original do GMA: o GMAalign. Esse novo programa possui como parâmetro um arquivo texto com os caminhos para cada um dos textos que formam os bitextos a serem alinhados e, opcionalmente, as etiquetas usadas para identificação do texto e das fronteiras de parágrafos e sentenças. A chamada ao método é feita pelo comando:

GMAalign <corpus paralelo> [<etiquetas>]

Com esse comando, o GMAalign irá executar basicamente três funções: pré-processamento dos textos, alinhamento e pós-processamento dos textos.

O pré-processamento dos textos tem o objetivo de formatá-los de acordo com o formato esperado pelo GMA e, para isso, as etiquetas de identificação do texto (<text></text>) e de fronteiras de parágrafos ( ) são removidas. No caso das fronteiras de sentenças (<s></s>), as finais (</s>) são removidas e as iniciais (<s>) são substituídas por marcadores de segmento (<EOS>) durante a geração dos eixos pelo programa **axis**. Além disso, a cada token (palavra ou símbolo separados por espaços) é atribuída a posição de seu caractere mediano (vide Figura 3). Um trecho do eixo gerado para um texto em PBr etiquetado é mostrado na Figura 5:

<text lang=pt id=art1R> <s>Neste artigo é apresentada uma ferramenta para validação e verificação de requisitos.</s><s>Essa ferramenta suporta a abordagem ERACE.</s><s>Tal abordagem parte do documento de requisitos do sistema e propõem a especificação das interações entre o sistema e seus agentes (cenários), e então os cenários são especificados detalhadamente.</s><s>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s> </text>	0 <EOS> 3 Neste 9.5 artigo 14 é 21 apresentada 29 uma 36.5 ferramenta 44.5 para 52 validação 58 e 65 verificação 72.5 de 79.5 requisitos 86 . 89 <EOS>

Figura 5 – Exemplo de um texto em PBr e parte do eixo gerado para ele.

Após o pré-processamento, os textos estão prontos para serem alinhados pelo método GMA. Nesse momento o programa principal, GMAalign, faz uma chamada ao programa de alinhamento original (GMA.csh) para cada um dos bitextos presentes no *corpus* paralelo passado como parâmetro. Por exemplo, para o par de textos paralelos art1R.txt e art1A.txt a chamada ao GMA.csh seria:

GMA.csh config art1R.txt art1A.txt > art1.txt

na qual o arquivo de configuração config contém os valores dos parâmetros do GMA otimizados para o PBr e o inglês (vide Subseção 4.2). Os textos paralelos art1R.txt e art1A.txt serão alinhados e a

saída será salva no arquivo art1.txt. O arquivo de saída gerado pelo GMA é motrado na Figura 6, onde o delimitador " <=> " separa os blocos alinhados de segmentos fonte e alvo.

```
1 <=> 1
2 <=> 2
3 <=> 3
4 <=> 4,5
```

Figura 6 – Exemplo de um arquivo de saída do GMA.

Quando vários segmentos estiverem envolvidos eles serão separados por vírgulas; e quando um dos lados for composto por um bloco vazio (nenhum segmento), ele será representado pela palavra *omitted*.

Uma representação geométrica dos alinhamentos da Figura 6 pode ser observada na Figura 7. Nessa figura está representado o mapeamento do bitexto (art1R-art1A) e as fronteiras das sentenças dos textos fonte e alvo. Os blocos alinhados são representados por linhas sólidas.

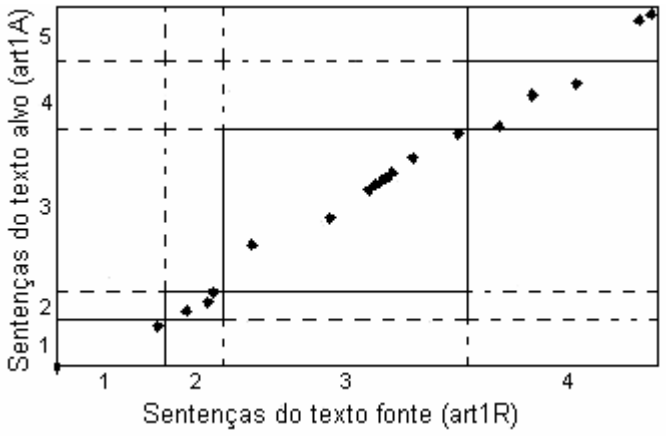

Figura 7 – Blocos alinhados do bitexto (art1R-art1A).

Como observado na Figura 6, a saída retornada pelo GMA não está no formato pré-estabelecido no projeto PESA para os arquivos alinhados e por isso precisa ser processada. Esse processamento é feito pela sub-rotina **saida** implementada no programa GMAalign. Essa sub-rotina lê cada linha do arquivo de saída (art1.txt, no exemplo anterior) e busca nos textos paralelos (art1R.txt e art1A.txt) as sentenças alinhadas. Dois novos arquivos são gerados com o mesmo nome e conteúdo dos textos fonte e alvo, porém com a extensão **.al** e os atributos id e corresp em suas etiquetas iniciais de sentença. Assim, a Figura 8 traz os textos art1R.txt e art1A.txt alinhados pelo GMA e pós-processados.

<text lang=pt id=art1R> <s id=art1R.1.s1 corresp=art1A.1.s1>Neste artigo é apresentada uma ferramenta para validação e verificação de requisitos.</s><s id=art1R.1.s2 corresp=art1A.1.s2>Essa ferramenta suporta a abordagem ERACE.</s><s id=art1R.1.s3 corresp=art1A.1.s3>Tal abordagem parte do documento de requisitos do sistema e propõem a especificação das interações entre o sistema e seus agentes (cenários), e então os cenários são especificados detalhadamente.</s><s id=art1R.1.s4 corresp='art1A.1.s4 art1A.1.s5'>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s> </text>	<text lang=en id=art1A> <s id=art1A.1.s1 corresp=art1R.1.s1>A tool to support requirements trading is presented.</s><s id=art1A.1.s2 corresp=art1R.1.s2>The tool supports the ERACE approach.</s><s id=art1A.1.s3 corresp=art1R.1.s3>This approach starts from the system's requirement document and proposes to specify interactions between the system and its agents (scenarios), and then the scenarios are specified in detail.</s><s id=art1A.1.s4 corresp=art1R.1.s4>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s id=art1A.1.s5 corresp=art1R.1.s4>An example to illustrates the approach is also presented.</s> </text>

Figura 8 – Exemplo de um bitexto alinhado pelo GMA.

Além desses três arquivos de saída gerados para cada bitexto alinhado, um outro é produzido pelo SIMR contendo o mapeamento do bitexto, ou seja, as posições das palavras consideradas pontos de correspondência nos dois textos. Após um processamento efetuado pelo programa **map_lista** esses valores numéricos são transformados em uma lista bilíngüe de palavras como mostrado na Figura 9.

suporta supports
documento document
interações interactions
agentes agents
cenários scenarios
cenários scenarios
heurísticas Heuristics
modelo model

Figura 9 – Exemplo de uma lista de pontos de correspondência gerada pelo SIMR.

A próxima subseção traz uma explicação sucinta do processo de otimização dos parâmetros citados nesta subseção, para o PBr e o inglês.

4.2 Otimização dos Parâmetros

Como visto na subseção anterior, o SIMR utiliza vários parâmetros para encontrar o mapeamento do bitexto, são eles: o tamanho fixo da cadeia e os limites para o LCSR, o nível de ambigüidade máximo da cadeia, a dispersão máxima do ponto e o desvio máximo do ângulo. Para garantir um bom desempenho do método é aconselhado re-otimizar esses parâmetros para cada novo par de línguas ou tipo de texto utilizado.

Os parâmetros citados foram otimizados seguindo o processo descrito em Melamed (1996). Nesse processo foram utilizados dois bitextos alinhados, com cerca de 500 sentenças cada, escritos em PBr e inglês. O primeiro bitexto, parte de um manual de PHP[5], foi submetido ao processo de re-otimização via *simulated annealing* (Vidal, 1993 *apud* Melamed, 2000). Essa técnica utiliza uma função que mede a diferença entre os pontos de correspondência verdadeiros e os mapeamentos do bitexto interpolados produzidos com os parâmetros atuais. Em termos geométricos, a diferença é uma distância e a métrica usada nesse caso é a raiz da distância média ao quadrado (*root mean squared distance* - RMS).

Os valores gerados nesse processo foram validados com o alinhamento do segundo bitexto, parte da constituição brasileira de 1988[6]. Os valores dos parâmetros obtidos nesse processo são apresentados na Tabela 2:

Tabela 2. Parâmetros do SIMR otimizados para o PBr e o inglês.

Parâmetros	PBr
Tamanho da cadeia	6
Limite mínimo para LCSR	4
Nível de ambigüidade máximo da cadeia	0,11
Dispersão máxima do ponto	15
Desvio máximo do ângulo	0,65

A escolha de textos pertencentes a domínios distintos baseou-se no texto em (Melamed, 1996) que afirma ser esta a situação ideal para a otimização dos parâmetros. Os *corpora* autêntico e pré-editado foram alinhados utilizando os parâmetros da Tabela 2 e avaliados apresentando os resultados descritos na próxima seção (Seção 5).

[5] Versão em inglês disponível em http://www.php.net/manual/en/ e versão em português disponível em http://www.php.net/manual/pt_BR/index.php (13/10/2002).
[6] Versão em inglês disponível em e http://www.georgetown.edu/pdba/Constitutions/Brazil/english98.html e versão em português disponível em http://www.georgetown.edu/pdba/Constitutions/Brazil/brazil88.html (13/10/2002).

5 Avaliação

Os métodos empíricos aqui apresentados foram selecionados para integrar o projeto PESA devido a diversos fatores, os principais são: apresentam uma precisão satisfatória, como demonstrado nas avaliações citadas nas seções 3 e 4; são representantes de métodos empíricos que utilizam critérios de alinhamento distintos; e são muito referenciados na literatura da área, inclusive como base para comparação de desempenho com outros métodos.

Para a avaliação desses métodos foram utilizados dois *corpora* de textos paralelos gerados como recursos lingüísticos no projeto PESA: o *corpus* autêntico e o *corpus* pré-editado[7]. Ambos são compostos por 65 pares de resumos e *abstracts* de trabalhos desenvolvidos no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, em São Carlos. A diferença entre esses dois *corpora* é que, no primeiro, os textos estão na forma como foram originalmente redigidos e, no segundo, os textos foram pré-editados por um tradutor humano para a remoção de erros do PBr e da tradução para o inglês.

Essa divisão visa enriquecer ainda mais a avaliação dos métodos no projeto PESA, uma vez que, segundo a literatura, esses métodos apresentam um melhor desempenho em *corpora* sem ruídos, ou seja, sem erros de qualquer espécie.

Os *corpora* autêntico e pré-editado foram ainda divididos em dois grupos: *corpora* de teste e *corpora* de referência. Os *corpora* de teste são compostos pelos textos dos *corpora* autêntico e pré-editado após um processo de etiquetação das fronteiras dos textos: texto completo, parágrafos e sentenças. Desse processo surgiram: o *corpus* autêntico de teste (CAT) e o *corpus* pré-editado de teste (CPT). Os *corpora* de referência, por sua vez, são compostos pelos mesmos textos dos *corpora* de teste já alinhados no nível sentencial e servem de base na comparação com os textos alinhados pelos métodos, já que são considerados corretos. Esses *corpora* receberam a denominação de: *corpus* autêntico de referência (CAR) e *corpus* pré-editado de referência (CPR)[8].

As métricas utilizadas para a avaliação dos métodos de alinhamento de textos paralelos são: *precision, recall* e *f-measure*, calculadas com base no alinhamento de referência. *Precision* é a porcentagem de alinhamentos corretos em relação a todos que foram propostos nos textos paralelos que compõem um *corpus*. *Recall* é a porcentagem de alinhamentos que foram propostos, entre todos os possíveis (no *corpus* de referência). E *F-measure* é a medida de freqüência, calculada como o dobro da razão entre o produto *recall* x *precision* e a soma *recall* + *precision* (Véronis e Langlais, 2000).

Dessa forma, *recall* indica a capacidade do método de alinhamento em encontrar as

[7] Mais detalhes em (Martins *et al*, 2001) disponível em: http://www.nilc.icmc.usp.br/nilc/projects/pesa.htm.
[8] Mais detalhes em (Caseli e Nunes, 2002) disponível em: http://www.nilc.icmc.usp.br/nilc/projects/pesa.htm.

correspondências. Já *precision* indica a capacidade do método de alinhamento em encontrar as correspondências corretas. Por fim, *F-measure* combina as duas anteriores em uma única métrica eficiente.

Portanto:

$$precision = \frac{NúmeroAlinhamentosCorretos}{NúmetoAlinhamentos\,\mathrm{Pr}\,opostos} \tag{6}$$

$$recall = \frac{NúmeroAlinhamentos\,\mathrm{Pr}\,opostos}{NúmeroAlinhamentos\,\mathrm{Re}\;ferência} \tag{7}$$

$$F = 2\frac{recall \times precision}{recall + precision} \tag{8}$$

Para efeito de avaliação do algoritmo em termos das três métricas anteriormente apresentadas, foram consideradas as médias dos valores calculados para todos os bitextos. Devido à limitação do método GC de só alinhar textos com o mesmo número de parágrafos, foram alinhados e avaliados apenas 64 pares de textos paralelos do *corpus* autêntico de teste (CAT) e 63 do *corpus* pré-editado de teste (CPT). As métricas calculadas para os *corpora* de teste comparados aos *corpora* de referência são apresentadas na Tabela 3.

Tabela 3. Métricas calculadas para os *corpora* alinhados pelo método GC.

	CAT	**CPT**
precision	0,6599	0,7348
recall	1,0918	1,0764
F	0,8226	0,8734

As mesmas métricas foram calculadas para os textos alinhados pelo método GMA, só que desta vez todos os 65 pares de textos do CAT e os 65 pares de textos do CPT foram alinhados e avaliados, uma vez que o método não apresenta a limitação de alinhar apenas textos com o mesmo número de parágrafos como o GC.

No caso do método GMA, realizaram-se duas avaliações distintas utilizando os parâmetros calculados para o português europeu e o inglês, fornecidos com o método; e para o PBr e o inglês, otimizados nesse trabalho. A Tabela 4 apresenta os valores desses parâmetros sob as denominações português europeu e português brasileiro (PBr), já que a outra língua (o inglês) é a mesma em ambos.

Tabela 4. Parâmetros para o português europeu e o brasileiro quando alinhados com o inglês.

Parâmetros	**Português Brasileiro**	**Português Europeu**
Tamanho da cadeia	6	8
Limite mínimo para LCSR	4	8
Nível de ambigüidade máximo da cadeia	0,11	0,26
Dispersão máxima do ponto	15	19
Desvio máximo do ângulo	0,65	0,66

A Tabela 5 apresenta os valores das métricas calculados para o CAT e o CPT alinhados pelo método GMA com os dois conjuntos de parâmetros da Tabela 4.

Tabela 5. Métricas calculadas para os *corpora* alinhados pelo GMA com os parâmetros da Tabela 4.

	Português Europeu		**Português Brasileiro**	
	CAT	**CPT**	**CAT**	**CPT**
precision	0,9610	0,9881	0,9578	0,9900
recall	0,9956	0,9990	0,9970	0,9990
F	0,9780	0,9935	0,9770	0,9945

A partir dos valores da Tabela 5 pode-se perceber que não há uma diferença significativa entre os valores calculados para os *corpora* alinhados com os parâmetros para o português europeu e o brasileiro, por isso optou-se pela utilização dos valores para o PBr, já que é este o idioma em estudo no projeto PESA. É importante ressaltar também que tanto no processo de alinhamento com os parâmetros do português europeu quanto com os do PBr foram usadas as *stoplists* para o PBr e o inglês, citadas na seção 4.1.

Após a avaliação dos *corpora* CAT e CPT alinhados pelos métodos GC e GMA, conclui-se que ambos apresentaram uma melhor precisão para o *corpus* pré-editado, confirmando o que já havia sido relatado na literatura: o desempenho dos métodos de alinhamento de textos paralelos é melhor em *corpus* sem ruídos.

Outra análise efetuada com os textos alinhados pelos métodos GC e GMA examinou quais são os tipos de alinhamentos mais freqüentes em cada um desses métodos comparados aos tipos de alinhamentos nos *corpora* de referência. Os resultados dessa análise foram divididos em duas tabelas uma vez que nem todos os 65 pares de textos dos *corpora* de teste foram alinhados pelo método GC e, dessa forma, os *corpora* de referência dos dois métodos são diferentes. Assim, a Tabela 6 apresenta os resultados para o método GC enquanto a Tabela 7, os do método GMA.

Tabela 6. Análise dos tipos de alinhamentos dos *corpora* alinhados pelo método GC.

	***Corpus* de Referência**		***Corpus* Alinhado pelo GC**	
Categoria	**Autêntico**	**Pré-editado**	**Autêntico**	**Pré-editado**
0:1 ou 1:0	6	2	45	35
1:1	348	381	352	352
2:1 ou 1:2	41	17	29	26
2:2	4	2	3	2
2:3 ou 3:2	1	-	-	-
Total	**400**	**402**	**429**	**415**

Tabela 7. Análise dos tipos de alinhamentos dos *corpora* alinhados pelo método GMA.

	***Corpus* de Referência**		***Corpus* Alinhado pelo GMA**	
Categoria	**Autêntico**	**Pré-editado**	**Autêntico**	**Pré-editado**
0:1 ou 1:0	6	2	6	2
1:1	352	394	360	399
2:1 ou 1:2	41	17	42	17
2:2	4	2	1	-
2:3 ou 3:2	1	-	-	-
Total	**404**	**415**	**409**	**418**

Embora em alguns casos os números apresentados nas tabelas anteriores para os *corpora* de referência e os *corpora* alinhados pelos métodos sejam bem próximos, nada se pode dizer sobre sua corretude. Para isso, é preciso avaliar os alinhamentos gerados pelos métodos analisando um caso não considerado pelas métricas *precision, recall* e *f-measure*: o alinhamento parcialmente correto.

Um exemplo de alinhamento parcialmente correto é mostrado na Tabela 9 e a versão (totalmente) correta do alinhamento é apresentada na Tabela 8.

Tabela 8. Sentenças do art1 do CAR.

PBr	**Inglês**
<s id=art1R.1.s4 corresp='art1A.1.s4 art1A.1.s5'>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s>	<s id=art1A.1.s4 corresp=art1R.1.s4>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s id=art1A.1.s5 corresp=art1R.1.s4>An example to illustrates the approach is also presented.</s>

Tabela 9. Sentenças do art1 do CAT alinhadas pelo método GC.

PBr	**Inglês**
<s id=art1R.1.s4 corresp=art1A.1.s4>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s>	<s id=art1A.1.s4 corresp=art1R.1.s4>Heuristics to evolve from the requirements model to the analysis are also presented.</s>
	<s id=art1A.1.s5 corresp=">An example to illustrates the approach is also presented.</s>

A partir da constatação dessa limitação, foi feita uma análise da quantidade de alinhamentos corretos – total e parcialmente – e errados nos *corpora* alinhados pelos métodos GC e GMA. O resultado dessa análise, em relação a cada tipo de alinhamento, é apresentado nas tabelas 10 e 11. A Tabela 12 resume as duas anteriores em uma classificação geral e comparativa dos alinhamentos produzidos pelos métodos.

Tabela 10. Classificação dos alinhamentos nos *corpora* alinhados pelo método GC.

	CAT			CPT		
Categoria	**Parcialmente**	**Corretos**	**Errados**	**Parcialmente**	**Corretos**	**Errados**
0:1	0	1	38	0	0	30
1:0	0	1	5	0	0	5
1:1	27	259	66	14	290	48
1:2	2	1	6	2	0	5
2:1	0	4	16	0	0	19
2:2	0	0	3	0	0	2
Total	**29**	**266**	**134**	**16**	**290**	**109**

Tabela 11. Classificação dos alinhamentos nos *corpora* alinhados pelo método GMA.

	CAT			CPT		
Categoria	**Parcialmente**	**Corretos**	**Errados**	**Parcialmente**	**Corretos**	**Errados**
0:1	0	0	2	-	-	-
1:0	-	1	3	0	1	1
1:1	3	349	8	1	396	2
1:2	0	31	2	0	16	0
2:1	0	7	2	0	1	0
2:2	0	1	0	-	-	-
Total	**3**	**389**	**17**	**1**	**414**	**3**

Tabela 12. Classificação geral dos alinhamentos nos *corpora* alinhados pelos métodos GC e GMA.

	***Corpus* Alinhado pelo GC**		***Corpus* Alinhado pelo GMA**	
Alinhamentos Propostos	**Autêntico**	**Pré-editado**	**Autêntico**	**Pré-editado**
Parcialmente Corretos	29 (6,76%)	16 (3,86%)	3 (0,73%)	1 (0,24%)
Totalmente Corretos	266 (62,00%)	290 (69,88%)	389 (95,11%)	414 (99,04%)
Errados	134 (31,24%)	109 (26,26%)	17 (4,16%)	3 (0,72%)
Total	**429**	**415**	**409**	**418**

Uma outra avaliação foi realizada com o intuito de comparar os métodos. Nela foi empregada uma metodologia de avaliação de algoritmos muito utilizada em Aprendizado de Máquina (AM), que utiliza o estimador *r-fold cross validation*. O texto que segue está fortemente baseado em (Freedman, Pisani e Purves, 1998 *apud* Baranauskas, 2001).

No *r-fold cross validation*, *n* exemplos são divididos aleatoriamente em *r* partições mutuamente exclusivas (*folds*) de tamanho aproximadamente igual a *n/r* exemplos cada. Um treinamento é efetuado com os exemplos contidos nos (r-1) *folds* e a hipótese induzida é testada no *fold* restante. Este processo é executado *r* vezes e, em cada uma delas, um *fold* diferente é usado para teste. O erro é calculado como a média dos erros obtidos em cada um dos *r folds*.

É importante ressaltar também que o *r-fold cross validation* é utilizado apenas para teste e os resultados não alteram de forma alguma o método avaliado. Embora ele seja usado em uma metodologia de avaliação de algoritmos para aprendizado de máquina, neste caso seus resultados são utilizados apenas na avaliação e não no aprendizado.

Dessa forma, dado um algoritmo A (um método de alinhamento) e um conjunto de exemplos T (um *corpus* de teste), T é dividido em *r* partições. Para cada partição *i*, uma hipótese *hi* é induzida e um erro *err(hi)*, i = 1, 2, ..., *r*, é calculado. Em seguida calcula-se a média, a variância e o desvio padrão para todas as partições através das equações (9), (10) e (11), respectivamente.

$$media(A) = \frac{1}{r}\sum_{i=1}^{r} err(hi) \tag{9}$$

$$\text{var}(A) = \frac{1}{r}\left[\frac{1}{r-1}\sum_{i=1}^{r}(err(hi) - media(A))^2\right] \tag{10}$$

$$sd(A) = \sqrt{\text{var}(A)} \tag{11}$$

O desvio padrão é uma medida de extrema importância na comparação de dois algoritmos, pois pode ser visto como uma medida da robustez do algoritmo. Essa robustez pode ser verificada calculando-se os erros sobre diferentes conjuntos de teste a partir de hipóteses induzidas utilizando diferentes conjuntos de treinamento. Se a diferença entre os erros calculados de um experimento para outro for muito grande, então o algoritmo não se comporta bem quando sujeito a mudanças no conjunto de treinamento, ou seja, não é robusto.

Os valores calculados segundo (9) e (11) para os métodos GC e GMA, considerando o valor de *f-measure* são mostrados na Tabela 13.

Tabela 13. Média e desvio padrão dos erros calculados para *f-measure* dos métodos GC e GMA.

	GC		**GMA**	
	CAT	**CPT**	**CAT**	**CPT**
media(A)	0,0541	0,0867	0,0136	0,0097
sd(A)	0,0162	0,0247	0,0015	0,0037

Para decidir se um algoritmo proposto – A_P – é melhor do que um algoritmo padrão – A_S – (com 95% de confiança) deve-se assumir o caso geral e determinar se a diferença entre os dois é significante ou não, assumindo uma distribuição normal (Weiss e Indurkhya, 1998 *apud* Baranauskas, 2001). Para isso, calcula-se a média e o desvio padrão combinados de acordo com as equações (12) e (13), respectivamente.

$$media(A_S - A_P) = media(A_S) - media(A_P) \tag{12}$$

$$sd(A_S - A_P) = \sqrt{\frac{sd(A_S)^2 + sd(A_P)^2}{2}} \tag{13}$$

Considerando os valores da Tabela 13, tem-se a média e o desvio padrão combinados para os métodos GC e GMA mostrados na Tabela 14:

Tabela 14. Média e desvio padrão combinados para GC e GMA.

	CAT	**CPT**
media(GC-GMA)	0,0541-0,0136=0,0405	0,0867-0,0097=0,0770
sd(GC-GMA)	0,0162-0,0015=0,0147	0,0247-0,0037=0,0210

O próximo passo é calcular a diferença absoluta, em desvios padrões, como mostrado na equação (14).

$$ad(A_S - A_P) = \frac{media(A_S - A_P)}{sd(A_S - A_P)} \quad (14)$$

Assim, se ad(A_S – A_P) > 0 então A_P é melhor do que A_S. Além disso, se ad(A_S – A_P) ≥ 2 desvios padrões então A_P supera A_S com 95% de confiança. Porém, se ad(A_S – A_P) ≤ 0 então A_S é melhor do que A_P e se ad(A_S – A_P) ≤ -2 pode-se dizer que A_S supera A_P com 95% de confiança.

Assim, no cálculo da diferença absoluta entre os métodos GC e GMA, foram obtidos os seguintes valores para os *corpora* autêntico e pré-editado, respectivamente:

$$ad(GC - GMA) = \frac{0,0405}{0,0147} = 2,7551$$

$$ad(GC - GMA) = \frac{0,0770}{0,0210} = 3,6667$$

A partir desses valores conclui-se que o GMA é melhor que o GC e, além disso, o GMA supera o GC com 95% de confiança.

As próximas subseções apresentam considerações sobre os dois métodos avaliados, ressaltando os casos de insucesso em ambos.

5.1 Considerações sobre o Método GC

A precisão apresentada pelo método GC nesse experimento não está dentro dos limites relatados para o alinhamento sentencial (acima de 95%), nem próxima à relatada pelos autores do método em (Gale e Church, 1991; Gale e Church, 1993) – 96%. Contudo, os valores calculados para *f-measure* para os *corpora* autêntico e pré-editado, 82,26% e 87,34%, respectivamente, são superiores aos relatados em (Véronis e Langlais, 2000): entre 62 e 82%.

A baixa precisão verificada nesse experimento para os textos alinhados pelo método GC pode ser explicada pela relação não tão forte entre os idiomas PBr e inglês (0,89 caracteres em inglês para cada caractere em PBr) como a apresentada para os pares Inglês-Alemão (1,1 caracteres

em alemão para cada caractere em inglês) e Inglês-Francês (1,06 caracteres em francês para cada caractere em inglês) em (Gale e Church, 1991; Gale e Church, 1993).

Além disso, o método GC apresenta a limitação de encontrar, apenas alinhamentos 1:1, 1:0, 0:1 e fusões de complexidade variável (m:n) com $0 \leq m, n \leq 2$. Essa limitação faz com que alinhamentos de blocos com mais de duas sentenças não sejam corretamente alinhados. Com relação aos tipos de alinhamentos mais freqüentes, notou-se que o método GC prioriza alinhamentos 1:1 e 0:1 ou 1:0 (Tabela 6), sendo esses últimos os campeões de erros (Tabela 10).

A seguir são apresentados alguns exemplos de casos de insucesso do método GC e os respectivos alinhamentos corretos do *corpus* de referência correspondente.

Exemplo 5.1.1: Um alinhamento 1:2 considerado como dois alinhamentos 1:1.

art10 do CAR

<s id=**art10R.1.s1** corresp='**art10A.1.s1 art10A.1.s2**'>O SPP2 (Servidor de Processamento Paralelo), desenvolvido no Laboratório de Computação de Alto Desempenho (LCAD-ICMC-USP) utiliza computadores convencionais conectados por uma rede de comunicação de alta velocidade.</s>	<s id=**art10A.1.s1** corresp=**art10R.1.s1**>Conventional computers connected by high-speed communication networks present a very low cost alternative to the MPPs (Massively Parallel Processors) for applications that demand high computing power.</s><s id=**art10A.1.s2** corresp=**art10R.1.s1**>The SPP2 (Parallel Processing Server), developed at the LCAD-ICMC-USP, is one of these systems.</s>

art10 do CAT após ser alinhado pelo método GC

<s id=**art10R.1.s1** corresp=**art10A.1.s1**>O SPP2 (Servidor de Processamento Paralelo), desenvolvido no Laboratório de Computação de Alto Desempenho (LCAD-ICMC-USP) utiliza computadores convencionais conectados por uma rede de comunicação de alta velocidade.</s><s id=**art10R.1.s2** corresp=**art10A.1.s2**>Pesquisadores da Universidade de Illinois desenvolveram uma camada de software de alto desempenho para a troca de mensagens entre máquinas conectadas por redes de alta velocidade Myrinet denominada Fast Messages, e que apresenta baixa latência na transmissão de mensagens e alta taxa de transferência.</s>	<s id=**art10A.1.s1** corresp=**art10R.1.s1**>Conventional computers connected by high-speed communication networks present a very low cost alternative to the MPPs (Massively Parallel Processors) for applications that demand high computing power.</s><s id=**art10A.1.s2** corresp=**art10R.1.s2**>The SPP2 (Parallel Processing Server), developed at the LCAD-ICMC-USP, is one of these systems.</s>

Exemplo 5.1.2: Dois alinhamentos 1:1 considerados como um alinhamento 2:2.

es6 do CAR

<s id=**es6R.1.s5** corresp=**es6A.1.s5**>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, a partir dos operadores implementados na ferramenta Proteum.</s><s id=**es6R.1.s6** corresp=**es6A.1.s6**>Procurando aplicar e validar o procedimento proposto, dois grupos distintos de programas são utilizados.</s>	<s id=**es6A.1.s5** corresp=**es6R.1.s5**>This work aims to investigate pragmatic alternatives for mutation analysis application and, in this context, a procedure for the determination of an essential mutant operators set for C is proposed, using Proteum testing tool.</s><s id=**es6A.1.s6** corresp=**es6R.1.s6**>Aiming to apply and validate the proposed procedure, two different groups of programs are used.</s>

es6 do CAT após ser alinhado pelo método GC

<s id=**es6R.1.s5** corresp='**es6A.1.s5 es6A.1.s6**'>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, a partir dos operadores implementados na ferramenta Proteum.</s><s id=**es6R.1.s6** corresp='**es6A.1.s5 es6A.1.s6**'>Procurando aplicar e validar o procedimento proposto, dois grupos distintos de programas são utilizados.</s>	<s id=**es6A.1.s5** corresp='**es6R.1.s5 es6R.1.s6**'>This work aims to investigate pragmatic alternatives for mutation analysis application and, in this context, a procedure for the determination of an essential mutant operators set for C is proposed, using Proteum testing tool.</s><s id=**es6A.1.s6** corresp='**es6R.1.s5 es6R.1.s6**'>Aiming to apply and validate the proposed procedure, two different groups of programs are used.</s>

Exemplo 5.1.3: Um alinhamento 1:2 considerado como um alinhamento 1:1 e um alinhamento 1:0.

art1 do CAR

<s id=**art1R.1.s4** corresp='**art1A.1.s4 art1A.1.s5**'>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s>	<s id=**art1A.1.s4** corresp=**art1R.1.s4**>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s id=**art1A.1.s5** corresp=**art1R.1.s4**>An example to illustrates the approach is also presented.</s>

art1 do CAT após ser alinhado pelo método GC

<s id=**art1R.1.s4** corresp=**art1A.1.s4**>Também são apresentadas heurísticas para a evolução do modelo de requisitos para modelos de análise, exemplificadas através do estudo de caso apresentado.</s>	<s id=**art1A.1.s4** corresp=**art1R.1.s4**>Heuristics to evolve from the requirements model to the analysis are also presented.</s><s id=**art1A.1.s5** corresp=**''**>An example to illustrates the approach is also presented.</s>

Exemplo 5.1.4: Um alinhamento 2:2 considerado como dois alinhamentos 1:1.

es12 do CAR

<s id=**es12R.3.s1** corresp='**es12A.3.s1 es12A.3.s2**'>Dessa forma, neste trabalho é apresentada uma ferramenta de injeção de defeitos de software, denominada ITool, baseada em um esquema de injeção de defeitos.</s><s id=**es12R.3.s2** corresp='**es12A.3.s1 es12A.3.s2**'>Esse esquema caracteriza o mapeamento de uma taxonomia de defeitos de software (Taxonomia de DeMillo) para os operadores de mutação do critério de teste Análise de Mutantes para a linguagem C.</s>	<s id=**es12A.3.s1** corresp='**es12R.3.s1 es12R.3.s2**'>In this perspective, in this work a software fault injection tool, named ITool, is presented.</s><s id=**es12A.3.s2** corresp='**es12R.3.s1 es12R.3.s2**'>This tool is based on a fault injection scheme that defines the mapping of a software fault taxonomy (DeMillo's Taxonomy) to the mutation operators of the Mutation Analysis criterion for C language.</s>

es12 do CAT após ser alinhado pelo método GC.

<s id=**es12R.3.s1** corresp=**es12A.3.s1**>Dessa forma, neste trabalho é apresentada uma ferramenta de injeção de defeitos de software, denominada ITool, baseada em um esquema de injeção de defeitos.</s><s id=**es12R.3.s2** corresp=**es12A.3.s2**>Esse esquema caracteriza o mapeamento de uma taxonomia de defeitos de software (Taxonomia de DeMillo) para os operadores de mutação do critério de teste Análise de Mutantes para a linguagem C.</s>	<s id=**es12A.3.s1** corresp=**es12R.3.s1**>In this perspective, in this work a software fault injection tool, named ITool, is presented.</s><s id=**es12A.3.s2** corresp=**es12R.3.s2**>This tool is based on a fault injection scheme that defines the mapping of a software fault taxonomy (DeMillo's Taxonomy) to the mutation operators of the Mutation Analysis criterion for C language.</s>

Com relação aos valores apresentados para *recall*, cabe aqui uma explicação de porque esses valores são maiores que 1. Tal fato está relacionado à maior prioridade que o método GC dá a alinhamentos do tipo 1:1 em detrimento dos outros tipos. No exemplo 5.1.4, duas sentenças no texto fonte (s_1 e s_2) e duas sentenças no texto alvo (t_1 e t_2) que no *corpus* de referência são alinhadas duas a duas ((s_1, s_2),(t_1, t_2)) foram alinhadas uma a uma ((s_1, t_1) e (s_2, t_2)) pelo método GC. Assim, no primeiro caso existe 1 alinhamento 2:2 e, no segundo, 2 alinhamentos 1:1.

5.2 Considerações sobre o Método GMA

O método GMA apresentou uma precisão dentro dos limites relatados na literatura – 94,2% (Véronis e Langlais, 2000) a 98,5% (Melamed, 2000) – e confirmou o fato de que os métodos de alinhamento de textos paralelos apresentam melhor precisão em *corpora* sem ruídos.

Com relação aos tipos de alinhamentos mais freqüentes, notou-se que o método GMA prioriza alinhamentos 1:1 e 1:2 ou 2:1 (Tabela 7), sendo os alinhamentos 1:0 e 0:1 os campeões de erros (Tabela 11).

Antes da apresentação de alguns exemplos de casos de insucesso do método GMA é interessante comentar o desempenho desse método no alinhamento dos exemplos apresentados na subseção anterior. O GMA alinhou corretamente os casos 5.1.2 e 5.1.3, e apresentou o mesmo resultado que o GC no exemplo 5.1.4. No caso do 5.1.1 o GMA, assim como o GC, também não

obteve sucesso, porém o alinhamento por ele gerado difere do alinhamento gerado pelo GC. Esse caso e outros de insucesso são apresentados a seguir.

Exemplo 5.2.1 (mesmo bitexto do 5.1.1): Um alinhamento 1:2 considerado como um alinhamento 0:1 e um 1:1.

art10 do CAR

<s id=**art10R.1.s1** corresp='**art10A.1.s1 art10A.1.s2**'>O SPP2 (Servidor de Processamento Paralelo), desenvolvido no Laboratório de Computação de Alto Desempenho (LCAD-ICMC-USP) utiliza computadores convencionais conectados por uma rede de comunicação de alta velocidade.</s>	<s id=**art10A.1.s1** corresp=**art10R.1.s1**>Conventional computers connected by high-speed communication networks present a very low cost alternative to the MPPs (Massively Parallel Processors) for applications that demand high computing power.</s><s id=**art10A.1.s2** corresp=**art10R.1.s1**>The SPP2 (Parallel Processing Server), developed at the LCAD-ICMC-USP, is one of these systems.</s>

art10 do CAT após ser alinhado pelo método GMA

<s id=**art10R.1.s1** corresp=**art10A.1.s2**>O SPP2 (Servidor de Processamento Paralelo), desenvolvido no Laboratório de Computação de Alto Desempenho (LCAD-ICMC-USP) utiliza computadores convencionais conectados por uma rede de comunicação de alta velocidade.</s>	<s id=**art10A.1.s1** corresp=**''**>Conventional computers connected by high-speed communication networks present a very low cost alternative to the MPPs (Massively Parallel Processors) for applications that demand high computing power.</s><s id=**art10A.1.s2** corresp=**art10R.1.s1**>The SPP2 (Parallel Processing Server), developed at the LCAD-ICMC-USP, is one of these systems.</s>

Exemplo 5.2.2: Um alinhamento 1:0 seguido de um alinhamento 1:1 considerado como um alinhamento 2:1.

bd1 do CAR

<s id=**bd1R.1.s2** corresp=**''**>Os dados são consultados (alimentados) livremente nas bases de dados de organizações independentes entre si, porém quando é necessária a troca de dados, como não existe uma previsão de integração, os dados somente podem ser trocados após uma preparação que impõem alguma forma de intervenção manual, construção de filtros especiais, etc., uma vez que a não existência de um esquema comum impede que os dados de uma base possa ser intercambiados com os de outra base.</s><s id=**bd1R.1.s3** corresp=**bd1A.1.s2**>No entanto, embora as bases de dados de diferentes organizações possam ser construídas de maneira totalmente independentes, a necessidade de uma troca significa que a semântica dos elementos manipulados, em particular daqueles que devem ser compartilhados é, no mínimo, semelhante.</s>	<s id=**bd1A.1.s2** corresp=**bd1R.1.s3**>Although the databases of different organizations can (and must) be totally built in an independent way, when some elements must be interchanged, the semantic of these elements are at least similar.</s>

bd1 do CAT após ser alinhado pelo GMA

<s id=**bd1R.1.s2** corresp=**bd1A.1.s2**>Os dados são consultados (alimentados) livremente nas bases de dados de organizações independentes entre si, porém quando é necessária a troca de dados, como não existe uma previsão de integração, os dados somente podem ser trocados após uma preparação que impõem alguma forma de intervenção manual, construção de filtros especiais, etc., uma vez que a não existência de um esquema comum impede que os dados de uma base possa ser intercambiados com os de outra base.</s><s id=**bd1R.1.s3** corresp=**bd1A.1.s2**>No entanto, embora as bases de dados de diferentes organizações possam ser construídas de maneira totalmente independentes, a necessidade de uma troca significa que a semântica dos elementos manipulados, em particular daqueles que devem ser compartilhados é, no mínimo, semelhante.</s>	<s id=**bd1A.1.s2** corresp='**bd1R.1.s2 bd1R.1.s3**'>Although the databases of different organizations can (and must) be totally built in an independent way, when some elements must be interchanged, the semantic of these elements are at least similar.</s>

Exemplo 5.2.3: Um alinhamento 1:0 considerado como um alinhamento 1:1.

cad1 do CAR

<s id=**cad1R.1.s4** corresp=''>Diversos critérios são propostos para a avaliação do sistema e para a determinação da sua adequação às principais aplicações na agricultura.</s>	

cad1 do CAT após ser alinhado pelo método GMA

<s id=**cad1R.1.s4** corresp=**cad1A.1.s4**>Diversos critérios são propostos para a avaliação do sistema e para a determinação da sua adequação às principais aplicações na agricultura.</s>	<s id=**cad1A.1.s4** corresp=**cad1R.1.s4**>A Type I system is further detailed and implemented, allowing for an evaluation of the technology.</s>

Exemplo 5.2.4: Um alinhamento 2:1 considerado como um alinhamento 1:0 seguido de um alinhamento 1:1.

sdpc2 do CAR

<s id=**sdpc2R.4.s1** corresp=**sdpc2A.4.s1**>Foram realizados estudos visando a validação e a avaliação da ferramenta.</s><s id=**sdpc2R.4.s2** corresp=**sdpc2A.4.s1**>Os resultados obtidos demonstram que a ferramenta possui comportamento estável e tem potencial para ser utilizada livremente em ambientes P.V.M. e M.P.I..</s>	<s id=**sdpc2A.4.s1** corresp='**sdpc2R.4.s1 sdpc2R.4.s2**'>The tool produced is tested by means of several examples which show a stable behaviour and that the tool can be easily used in both P.V.M. and M.P.I. environments.</s>

sdpc2 do CAT após ser alinhado pelo método GMA

<s id=**sdpc2R.4.s1** corresp="">Foram realizados estudos visando a validação e a avaliação da ferramenta.</s><s id=**sdpc2R.4.s2** corresp=**sdpc2A.4.s1**>Os resultados obtidos demonstram que a ferramenta possui comportamento estável e tem potencial para ser utilizada livremente em ambientes P.V.M. e M.P.I..</s>	<s id=**sdpc2A.4.s1** corresp=**sdpc2R.4.s2**>The tool produced is tested by means of several examples which show a stable behaviour and that the tool can be easily used in both P.V.M. and M.P.I. environments.</s>

Além dos textos de entrada alinhados, o GMA produz como co-produto um outro recurso lingüístico: uma lista bilíngüe de palavras consideradas pontos de correspondência durante o processo de mapeamento do bitexto.

6 Conclusões e Trabalho Futuro

O método empírico de alinhamento sentencial de textos paralelos proposto por Gale e Church (1991, 1993), também referenciado como GC neste texto, utiliza um modelo estatístico simples que leva em consideração apenas o tamanho das sentenças, em caracteres, para determinar as correspondências entre elas. Esse método é um dos mais importantes e também um dos mais referenciados da área devido à sua simplicidade e a uma precisão satisfatória relatada nos primeiros experimentos.

Porém, o método GC apresenta algumas limitações como a de só alinhar textos com números iguais de parágrafos ou a de limitar os tipos de alinhamentos em *m*:*n* com $0 \leq m, n \leq 2$. Como trabalho futuro, poderão ser estudadas algumas melhorias no código que eliminariam as limitações acima citadas com o objetivo de aumentar a precisão do método e torná-lo mais eficiente. Também poderá ser estudada a possibilidade de incluir algum tipo de informação lingüística no processo de alinhamento do método, tornando-o um método híbrido.

O segundo método estudado neste trabalho, o método GMA, também é um método empírico e utiliza a técnica de reconhecimento de padrão para determinar os pontos de correspondência entre as sentenças a serem alinhadas (Melamed, 2000). Mesmo sem utilizar recursos lingüísticos adicionais no processo de alinhamento, o método apresentou um desempenho muito bom comparado ao método GC e aos relatos da literatura.

Pretende-se ainda estudar o desempenho deste método com a utilização de uma lista de palavras âncoras para auxiliar o processo de alinhamento, que neste experimento baseou-se apenas em cognatos. Com esse incremento o método se tornará híbrido e seu desempenho será comparado ao empírico aqui estudado.

A partir dos resultados obtidos na avaliação dos métodos GC e GMA para os *corpora* de resumos e *abstracts* no domínio da computação, pode-se afirmar que o método GMA obteve melhor desempenho com 95% de confiança.

Assim, os métodos GC e GMA formam a primeira das três classes de métodos a ser estudada no projeto PESA: a dos métodos empíricos. A partir de agora serão estudados, implementados e avaliados os representantes das demais classes (lingüísticos e híbridos) referenciando sempre os resultados aqui relatados para permitir a comparação entre eles e, possivelmente, a determinação de um método que tenha apresentado uma precisão satisfatória no alinhamento sentencial de textos paralelos PBr e inglês.

7 Referências Bilbiográficas

Baranauskas, J. A. (2001). Extração automática de conhecimento por múltiplos indutores. Tese de Doutorado. ICMC-USP, São Carlos.

Caseli, H. M. (2002). Alinhamento Sentencial de Textos Paralelos Português-Inglês. Monografia de Qualificação. ICMC-USP, São Carlos, Março.

Caseli, H.M.; Nunes, M.G.V. (2002). *A construção dos recursos lingüísticos do projeto PESA.* Série de Relatórios do NILC. NILC-TR-02-07, Junho 2002.

Caseli, H.M.; Feltrim, V.D.; Nunes, M.G.V. (2002). *TagAlign: Uma ferramenta de pré-processamento de textos*. Série de Relatórios do NILC. NILC-TR-02-09, Junho.

Campbell, J.A.; Chatterjee, N.; Dawkins, N. (1998). Experiments in Automated Alignment of Text over Several Languages. *In:* Proceedings of the International Conference on Computational Linguistics, Speech and Document Processing. Indian Statistical Institute, Calcutta, p. C-47 - C-54.

Freedman, D.; Pisani, R.; Purves, R. (Eds.) (1998). Statistics (Third ed.). W. W. Norton & Company.

Gale, W. A.; Church, K. W. (1991). A program for aligning sentences in bilingual corpora. *In*: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL), Berkley, p.177-84.

Gale, W. A.; Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *In*: Computational Linguistics, 19 (3), p.75-102.

Kay, M.; Röscheisen, M. (1988). Text-translation alignment. Technical Report. Xerox Palo Alto Research Center.

Kay, M.; Röscheisen, M. (1993). Text-translation alignment. *Computational Linguistics*, 19 (1), p.121-42.

Martins, M. S.; Caseli, H. M.; Nunes, M. G. V.(2001). A construção de um corpus de textos paralelos inglês-português. NILC-TR-01-05.

Melamed, I. D. (1996). Porting SIMR to New Language Pairs, IRCS Technical Report #96-26.

Melamed, I. D. (2000). Pattern recognition for mapping bitext correspondence. *In*: Véronis, J., ed. *Parallel text processing: Alignment and use of translation corpora*. s.l.: Kluwer Academic Publishers, p.25-47.

Weiss, S. M.; Indurkhya, N. (1998). Predictive Data Mining: A Practical Guide. San Francisco, CA: Morgan Kaufmann.

Véronis, J.; Langlais, P. (2000). Evaluation of parallel text alignment systems: The ARCADE Project. *In*: Véronis, J., ed. *Parallel text processing: Alignment and use of translation corpora.* s.l.: Kluwer Academic Publishers, p.369-88.

Vidal, R. V. V. (1993) (Ed). *Applied Simulated Annealing*. Heidelberg: Springer-Verlag.