



Trabalho Prático 0 - *Warm up*

Valor: 0,5 pontos extras (5% da nota total)

Data de Entrega: 30/03/2010

*Especificação do Problema: Auxiliar de Redação*¹

A publicação dos resultados de pesquisa é uma parte muito importante de todo o trabalho científico. Normalmente a publicação desses resultados ocorre na forma de artigos em congressos e periódicos depois de serem revisados quanto à forma e conteúdo. Infelizmente, na academia é comum encontrar artigos difíceis de ler e com erros de ortografia e de gramática.

Para ajudar na parte específica da forma, existem algumas ferramentas que podem nos auxiliar. Por exemplo, no Unix existe um utilitário chamado `ispell` que faz a verificação da grafia (*spelling checker*) de textos escritos em inglês. Para outros ambientes e línguas existem ferramentas específicas. Existem também ferramentas comerciais baseadas em técnicas de inteligência artificial que tentam descobrir erros gramaticais, estilo literário e outras informações como estatísticas sobre parágrafos.

O objetivo deste exercício de programação é especificar, projetar e implementar um sistema que o ajude no processo de redação. Esta é uma tarefa que exige um projeto cuidadoso das estruturas de dados e de seus respectivos algoritmos para poderem ser utilizados de forma eficiente na prática.

Considerações a serem feitas

- Este programa deveria ler um arquivo contendo um texto sem nenhuma formatação (“arquivo ASCII”) onde cada sentença termina por um sinal de pontuação (“.”, “?”, “!”);
- Cada parágrafo é separado por pelo menos uma linha em branco;
- Desconsidere na análise a ser feita pelo seu programa “stop words” como artigos (a, o, as, os) e conjunções (e, ou) que não possuem conteúdo semântico. A lista de “stop words” deve ser fornecida através de um arquivo a ser criado pelo aluno (você também deve especificar o formato desse arquivo);

¹A especificação deste trabalho foi extraída de [1] e então adaptada.

- A saída do seu programa irá fazer referência ao arquivo de entrada. Considere a primeira linha do arquivo de entrada como “1”, a segunda como “2” e assim por diante, para fazer essa referência;
- Uma palavra é uma sequência de letras delimitada por branco e símbolos de pontuação;
- Faz parte do projeto do seu programa fornecer uma saída legível;
- Os arquivos a serem usados para teste serão disponibilizados na página da disciplina;
- Outras suposições podem ser feitas desde que devidamente justificadas e não alterem as que foram descritas acima.

Saída

Este programa deverá fornecer as seguintes informações:

1. Para cada parágrafo do texto:

- (a) O número da linha no arquivo fonte onde começa;
- (b) O número de sentenças no parágrafo;
- (c) O número de palavras de cada sentença incluindo e não incluindo “stop words”.

Razão para incluir esta informação: Um problema que ocorre na redação são parágrafos formados por uma única sentença (o que deve ser evitado), sentenças muito longas ou curtas, ou mesmo uma mistura dessas situações.

2. Para cada palavra que não é “stop word”:

- (a) Cada palavra listada em ordem alfabética com a frequência de sua ocorrência no texto indicando as linhas no arquivo fonte onde aparecem;
- (b) No caso de uma palavra aparecer mais de uma vez no mesmo parágrafo indicar também a sua ocorrência no parágrafo e as distâncias entre elas incluindo “stop words”.

Razão para incluir esta informação: é comum repetir a mesma palavra várias vezes ao longo do texto ou num mesmo parágrafo. Os exemplos a seguir foram tirados de textos técnicos publicados na literatura:

- ... acerca de alimentação, moradia, saneamento, etc., acerca das condições de vida e trabalho. ... sobre nascimentos, doenças transmissíveis, etc. ... população por sexo, idade, procedência, etc. ... água, saneamento básico, recolhimento de lixo, etc. ... escolaridade, renda média, etc. ... ambulatórios, hospitais, etc. ... categoria profissional, lotação, etc. ... consultas, exames, vacinas, etc. ... custos, gastos, fontes financiadoras, etc.

[Normalmente num trabalho técnico deve-se evitar o uso de “etc” e ser mais claro enumerando ou listando os itens. Neste trecho “etc” aparece nove vezes.]

- Uma das tarefas mais importantes de todo o trabalho científico é a publicação dos resultados do trabalho de pesquisa.

[Palavra “trabalho” repetida duas vezes na mesma sentença separada por apenas sete vocábulos.]

3. Ocorrência de certas expressões sendo que para cada expressão deve ser indicado a linha no arquivo fonte onde aparece. O conjunto de expressões a serem analisadas deve ser lido de um arquivo a ser criado pelo aluno (você também deve especificar o formato desse arquivo).

Razão para incluir esta informação: Existem certas expressões ou palavras em português que são frequentemente usadas mas não existem. O exemplo mais comum é “em nível de” ou “a nível de” como é bem discutido pelo filólogo Vianney Mesquita, professor da Universidade Federal do Ceará em um de seus livros.

Também temos exemplos de palavras que não existem em português ou são usadas com sentidos inadequados como “deletar” no sentido de remover, e “suportar” no sentido de ter uma funcionalidade. Os exemplos a seguir foram tirados de textos técnicos publicados na literatura:

- Garantir maior disponibilidade dos sistemas, na medida em que problemas locais (em nível de hardware, software) não afetam o conjunto.
- Este trabalho analisou apenas alguns órgãos responsáveis pela segurança pública a nível do Estado de Minas Gerais e a Polícia Federal.

O que deve ser entregue

- Código fonte do programa em C ou C++ (bem indentado e comentado).
- Documentação do trabalho.

Entre outras coisas, a documentação deve conter:

1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Muito importante: os códigos utilizados nas implementações devem ser inseridos na documentação.
3. Análise de Complexidade: estudo da complexidade de tempo e espaço das funções implementadas e do programa como um todo (notação O).
4. Listagem de testes executados: os testes executados devem ser apresentados e analisados e discutidos, quando convier.

5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sítio da Internet se for o caso. Uma referência bibliográfica deve ser citada no texto quando da sua utilização.
7. Em $\text{\LaTeX}$: A documentação deve ser elaborada obrigatoriamente em $\text{\LaTeX}$. Veja modelo de como fazer o trabalho em latex: <http://www.decom.ufop.br/menotti/paa101/tps/modelo.zip>
8. Formato final: mandatoriamente em PDF (<http://www.pdf995.com/>).

Como deve ser feita a entrega

A entrega DEVE ser feita via Moodle (www.decom.ufop.br/moodle) na forma de um único arquivo zipado, contendo o código fonte, arquivos diversos e a documentação. Também deve ser entregue a documentação impressa na próxima aula teórica após a data de entrega do trabalho.

Comentários Gerais

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, identificação e comentários no programa também serão avaliados;
- O trabalho é individual (grupo de UM aluno);
- Trabalhos copiados (e FONTE) terão nota zero. Devido a recorrentes problemas com cópias de trabalhos (plágios), os autores de trabalhos copiados também terão a maior nota dentre os testes teóricos levada a zero, como forma de punição e coação ao plágio acadêmico;
- Trabalhos entregues em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero;
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.

Referências

- [1] Antônio Alfredo Ferreira Loureiro. Trabalho prático de aquecimento da disciplina de Projeto e Análise de Algoritmos do PPGCC/UFMG - 1o. semestre de 2010, 2010. http://homepages.dcc.ufmg.br/~loureiro/alg/101/TP_Aquecimento.pdf, visitado em 17/03/2010.