Universidade Federal de Mato Grosso do Sul
Campus de Três Lagoas
Bacharelado em Sistemas de Informação
Algoritmos e Programação II

Segunda Avaliação

Análise Textual

1 Descrição

O **Processamento de Linguagem Natural** (PLN) é uma área de pesquisa interdisciplinar que abrange, principalmente, Linguística e Inteligência Artificial com o objetivo de desenvolver programas para que os computadores possam executar tarefas úteis que envolvem a linguagem humana, como a comunicação homem-máquina, ou mesmo a comunicação entre humanos por meio do processamento de texto ou discurso.

Em PLN, *Part-of-Speech Tagging* (marcação ou etiquetagem PoS), também chamada de marcação de classe gramatical, consiste em etiquetar gramaticalmente elementos textuais na identificação de palavras como substantivos, verbos, adjetivos, advérbios, etc. PoS Tagger são utilizados em diversas aplicações de PLN tais como, tradutores automáticos, revisores gramaticais, sistemas de apoio à escrita e reconhecimento da fala. Devido a este fato, existem muitos trabalhos nessa área.

O objetivo deste trabalho é analisar um texto com a marcação POS e gerar um relatório com alguns dados que serão utilizados para futuras análises de textos.

1.1 *Part-of-Speech* (PoS)

Os textos com as palavras e suas respectivas marcações PoS são utilizados para construção de diversas aplicações. Existem diversos algoritmos para realizar a marcação PoS e que utilizam diferentes técnicas. Neste trabalho, iremos utilizar os textos anotados pela ferramenta do grupo de pesquisa de PLN da Stanford University, disponível em `http://nlp.stanford.edu:8080/parser/index.jsp`. As palavras do texto são classificadas de acordo com tags (etiquetas) do conjunto de Penn Treebank, conforme Tabela 1.

Considere o seguinte texto original:

```
My dog also likes eating sausage.
```

Após a marcação PoS teremos:

```
My/PRP$ dog/NN also/RB likes/VBZ eating/VBG sausage/NN ./.
```

Tabela 1: Tabela Símbolos Marcação PoS

Identificador	PoS	Classe Gramatical
1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential there
5	FW	Foreign word
6	IN	Preposition/subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PRP$	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle
24	SYM	Symbol (mathematical or scientific)
25	TO	to
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund/present
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd ps. sing. present
32	VBZ	Verb, 3rd ps. sing present
33	WDT	wh-determiner
34	WP	wh-pronoun
35	WP$	Possessive wh-pronoun
36	WRB	wh-adverb
37	#	Pound sign
38	$	Dollar sign
39	.	Sentence-final punctuation
40	,	Comma

Identificador	PoS	Classe Gramatical
41	:	Colon, semi-colon
42	(	Left bracket character
43	)	Right bracket character
44	´´	Straight double quote
45	‘	Left open single quote
46	“	Left open double quote
47	’	Right close single quote
48	”	Right close double quote

Na Tabela 1, veja os símbolos dos itens de 44 a 48 e observe a diferença sutil. Aspas curvas são as aspas usadas em uma boa tipografia. Existem quatro caracteres de aspas curvas: as aspas simples de abertura (‘), as aspas simples de fechamento (’), as aspas duplas de abertura (“) e as aspas duplas de fechamento (”). As aspas retas (´´) são as aspas genéricas localizadas perto da tecla de retorno.

2 Implementação

A implementação deste trabalho consiste em ler um arquivo com um texto e suas respectivas marcações PoS, como descrito na Seção 1.1, e gerar um relatório que contenha informações de totais de palavras, frequências de palavras e frequência por classes gramaticais, de acordo com a Tabela 1.

O seu programa tem como entrada um arquivo texto com as palavras e suas respectivas marcações PoS, como apresentado no exemplo da Seção 1.1. Esse arquivo deve ser informado para seu programa via argumento. Ou seja, o nome completo do arquivo deve ser passado para seu programa antes do início da execução. Por exemplo, suponha que o arquivo de texto esteja salvo com o nome `texto01.pos`, e que seu executável esteja nomeado como `analisa`. A chamada de execução do seu programa deve ser realizada seguinte maneira:

```
user@alg2:.$./analisa texto01.pos
```

O seu programa deve gerar um arquivo de saída com o mesmo nome do arquivo de entrada, alterando a extensão de “`.pos`” para “`.csv`”.

2.1 Arquivos de Entrada e de Saída

O arquivo de entrada é um arquivo no formato texto que contém as palavras do texto original, com letras maiúsculas e minúsculas, seguidas do caractere **‘/’** e seguido da sua respectiva marcação PoS. Na Figura 1 é apresentado um texto original sem marcações, e o seu respectivo arquivo de entrada após a marcação PoS é apresentado na Figura 2.

Palavras compostas são formadas por palavras independentes, ligadas por hífen, que juntas possuem um novo significado, por exemplo: *broad-coverage*, *natural-language* e *state-of-the-art*. Para facilitar o processamento do programa de marcação PoS e do

seu trabalho, o símbolo **hífen** (‘-’) entre as palavras foi substituído pelo ***underscore*** (‘_’). Desse modo, as palavras do exemplo a seguir são escritas como *broad_coverage*, *natural_language* e *state_of_the_art*.

Our research has resulted in state_of_the_art technology for robust, broad_coverage natural_language processing/NN in a number of languages. We provide a widely used, integrated NLP toolkit, Stanford CoreNLP.

Figura 1: Exemplo de arquivo com o texto original.

Our/PRP$ research/NN has/VBZ resulted/VBN in/IN state_of_the_art/JJ technology/NN for/IN robust/JJ ,/, broad_coverage/JJ natural_language/NN processing/NN in/IN a/DT number/NN of/IN languages/NNS ./. We/PRP provide/VBP a/DT widely/RB used/VBN ,/, integrated/VBN NLP/NN toolkit/NN ,/, Stanford/NNP CoreNLP/NNP ./.

Figura 2: Exemplo de arquivo de entrada.

O arquivo de saída deve conter as seguintes informações em cada linha:

- total de palavras lidas;
- total de palavras distintas;
- todas marcações PoS e suas respectivas frequências em ordem alfabética; e,
- todas palavras e suas respectivas frequências em ordem alfabética.

Na Figura 3 é apresentado um exemplo de arquivo de saída, na primeira linha do arquivo deve conter o “`TOTAL DE PALAVRAS, `” seguido do respectivo valor, na segunda linha, o “`TOTAL DE PALAVRAS DISTINTAS, `” seguido do respectivo valor, a linha seguinte deve estar em branco para iniciar o resultado da análise das marcações PoS e na quarta linha apresentar o texto “`PoS, FREQ`”. Na sequência, em cada linha, haverá uma marcação PoS seguida por vírgula (‘`,`’), um espaço em branco e sua frequência no texto. Quando esta parte terminar, a linha seguinte deve estar em branco e a próxima deve apresentar o texto“`PALAVRA, FREQ`”. Posteriormente, em cada linha, haverá uma palavra seguida por vírgula (‘`,`’), um espaço em branco e sua frequência no texto, até que todas as palavras distintas sejam apresentadas.

No arquivo de saída, as palavras que indicam as marcações PoS devem estar em letras maiúsculas e as palavras do texto devem estar em letras minúsculas. Os caracteres da Tabela 1 que indicam as marcações de 37 à 48, inclusive, devem ser desconsideradas nesta implementação, e não podem estar no arquivo de saída.

É muito **IMPORTANTE** que o seu arquivo de saída seja exatamente igual ao formato exigido, caso contrário, poderá ocorrer erros nas próximas etapas. Uma parte da avaliação será realizada por testes automatizados e se seu arquivo de saída não apresentar o formato solicitado serão descontados pontos da sua nota. Os trabalhos que não respeitarem os padrões definidos receberão nota ZERO.

```
TOTAL DE PALAVRAS, 27
TOTAL DE PALAVRAS DISTINTAS, 25

PoS, FREQ
DT, 2
IN, 4
JJ, 3
NN, 7
NNP, 2
NNS, 1
PRP, 1
PRP$, 1
RB, 1
VBN, 3
VBP, 1
VBZ, 1

PALAVRA, FREQ
a, 2
broad_coverage, 1
corenlp, 1
for, 1
has, 1
in, 2
integrated, 1
languages, 1
natura_language, 1
nlp, 1
number, 1
of, 1
our, 1
processing, 1
provide, 1
research, 1
resulted, 1
robust, 1
stanford, 1
state_of_the_art, 1
technology, 1
toolkit, 1
used, 1
we, 1
widely, 1
```

Figura 3: Exemplo de arquivo de saída para o arquivo de entrada da Figura 2.

3 Relatório

Além do programa para analise do texto, como detalhado na seção anterior, você também deverá entregar um relatório da implementação. Esperamos que você descreva no seu relatório a sua solução em alto nível, ou seja, descreva os principais passos da sua

solução sem entrar no nível de código. É interessante também que o relatório informe como estão organizados os dados do seu programa e as principais estruturas de dados (pilha, fila, lista, etc.) que você utilizou, como as utilizou e qual o impacto no tempo de execução da sua solução. O seu relatório deve conter uma seção para explicar o tempo de execução da sua solução. Não é necessário fazer a demonstração, mas você deve apresentar avaliar as principais funções.

Pense no relatório como uma chance de explicar para os professores as decisões que você tomou durante a implementação do seu programa e os problemas que foram enfrentados nesse período. Por isso, é importante também que você informe as principais dificuldades encontradas e as maneiras adotadas para tentar superar tais dificuldades. O seu relatório deve demonstrar que você tem domínio sobre o que foi desenvolvido, mesmo que, por algum motivo, não tenha obtido o sucesso desejado.

Ao final do relatório, faça uma auto-avaliação sobre sua participação no desenvolvimento do trabalho. Finalize a auto-avaliação com uma nota entre zero e dez, inclusive. Em caso de trabalhos realizados em grupo, faça também uma breve avaliação dos seus colegas, incluindo também uma nota para eles. Fique tranquilo, as informações colocadas nos relatórios não serão divulgadas.

O relatório tem um peso importante na nota final do trabalho. Seja organizado e claro na elaboração, sempre tomando cuidado com erros gramaticais e de concordância.

4 Entrega

4.1 Codificação e Execução

O código-fonte final entregue, será compilado com os seguintes parâmetros:

```
gcc trab1.c -o trab1.exe -ansi -pedantic -Wall
```

Caso o código-fonte não compile com este comando, o trabalho não será considerado e receberá nota ZERO. É permitido o uso das seguintes bibliotecas: `stdio.h`, `stdlib.h`, `string.h` e `ctype.h`.

É proibido o uso de qualquer outra biblioteca. Caso utilize qualquer outra biblioteca, além das citadas, o trabalho não será considerado e receberá nota ZERO.

4.2 Estruturas de dados

O programa deve utilizar `listas encadeadas dinamicamente` e caso seja necessário, outras estruturas de dados já estudadas até o momento. O programa não pode conter variáveis globais. Além disso, o programa deve ser bem comentado e deve utilizar a modularização de modo eficiente.

4.3 Formação dos grupos e datas

O trabalho pode ser realizado em grupos com, no máximo, dois integrantes (duplas). Porém, os relatórios devem ser INDIVIDUAIS. Por mais que a implementação tenha sido

feita em conjunto, as dificuldades encontradas e o modo de descrever a solução em alto nível muda de pessoa para pessoa. Portanto, evite cópias nos relatórios, mesmo que parciais. Relatórios de um trabalho que sejam considerados muito semelhantes terão penalidades na nota atribuída.

4.4 Forma de Entrega

O trabalho deverá ser entregue no Ambiente Virtual de Aprendizagem da UFMS (AVA) **até as 23:59 do dia 19/11/2021**. Serão disponibilizados dois links para entrega das atividades: um para o código-fonte na linguagem C e outro para os relatórios. Caso o trabalho seja realizado em grupo, apenas um dos integrantes deve encaminhar o código-fonte no primeiro link.

Encerrado o prazo, não serão mais aceitos trabalhos. Portanto, não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

4.5 Arquivo com o programa fonte

Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Lembrem-se que um programa bem descrito e bem organizado é a chave de seu sucesso.

5 Critérios de Avaliação

A nota atribuída a este trabalho será composta por outras duas notas: programa e relatório. A nota atribuída ao programa levará em consideração as respostas geradas para cada instância de teste e a organização e corretude do código-fonte. Já para a nota do relatório, será considerado, além do uso correto do português, o domínio apresentado pelo acadêmico ao descrever sua solução.

Ademais, cada uma das notas descritas acima receberá uma nota entre zero e dez. A nota final desse trabalho será dada pela fórmula:

$$A2 = 0.6 \times NP + 0.4 \times NR,$$

onde NP e NR são as notas obtidas no programa e no relatório, respectivamente.

6 Exemplo completo dos tipos de arquivo

Na Figura 4 é apresentado um arquivo de exemplo entrada que poderá ser testado pelo seu programa. O arquivo resultante do seu programa deve ser igual ao conteúdo, e padrões, das Figura 5, 6 e 7.

```
The/DT Natural/NNP Language/NNP Processing/NNP Group/NNP at/IN Stanford/NNP University/NNP is/VBZ a/DT team/NN of/IN faculty/NN ,/, postdocs/NNS ,/, programmers/NNS and/CC students/NNS who/WP work/VBP together/RB on/IN algorithms/NNS that/WDT allow/VBP computers/NNS to/TO process/VB ,/, generate/VB ,/, and/CC understand/VB human/JJ languages/NNS ./.
Our/PRP$ work/NN ranges/VBZ from/IN basic/JJ research/NN in/IN computational/JJ linguistics/NNS to/IN key/JJ applications/NNS in/IN human/JJ language/NN technology/NN ,/, and/CC covers/VBZ areas/NNS such/JJ as/IN sentence/NN understanding/NN ,/, automatic/JJ question/NN answering/NN ,/, machine/NN translation/NN ,/, syntactic/JJ parsing/NN and/CC tagging/NN ,/, sentiment/NN analysis/NN ,/, dialogue/NN agents/NNS ,/, and/CC models/NNS of/IN text/NN and/CC visual/JJ scenes/NNS ,/, as/RB well/RB as/IN applications/NNS of/IN natural/JJ language/NN processing/NN to/IN the/DT digital/JJ humanities/NNS and/CC computational/JJ social/JJ sciences/NNS ./.
```

Figura 4: Arquivo com um texto em inglês e suas respectivas marcações PoS.

```
TOTAL DE PALAVRAS, 91
TOTAL DE PALAVRAS DISTINTAS, 69

PoS, FREQ
CC, 7
DT, 3
IN, 12
JJ, 13
NN, 20
NNP, 6
NNS, 15
PRP, 1
RB, 3
TO, 1
VB, 3
VBP, 2
VBZ, 3
WDT, 1
WP, 1

PALAVRA, FREQ
a, 1
agents, 1
algorithms, 1
allow, 1
```

Figura 5: Exemplo de arquivo de saída.

```
analysis, 1
and, 7
answering, 1
applications, 2
areas, 1
as, 3
at, 1
automatic, 1
basic, 1
computational, 2
computers, 1
covers, 1
dialogue, 1
digital, 1
faculty, 1
from, 1
generate, 1
group, 1
human, 2
humanities, 1
in, 2
is, 1
key, 1
language, 3
languages, 1
linguistics, 1
machine, 1
moodels, 1
natural, 2
of, 3
on, 1
our, 1
parsing, 1
postdocs, 1
process, 1
processing, 2
programmers, 1
question, 1
ranges, 1
research, 1
scenes, 1
sciences, 1
sentence, 1
sentiment, 1
social, 1
stanford, 1
students, 1
such, 1
syntactic, 1
tagging, 1
team, 1
technology, 1
text, 1
that, 1
```

Figura 6: Continuação do arquivo de saída para o arquivo da Figura 5.

```
the, 2
to, 3
together, 1
translation, 1
understand, 1
understanding, 1
university, 1
visual, 1
well, 1
who, 1
work, 2
```

Figura 7: Continuação do arquivo de saída para o arquivo da Figura 5.