

Relatório do trabalho I

Mensurador de sentimentos em reviews

Rafael F. T. de Melo
Universidade Tecnológica
Federal do Paraná
Ponta Grossa, Paraná
Email: rafaelmelo.1999@alunos.utfpr.edu.br

Resumo—The abstract goes here.

1. Introdução

Um dos grandes destaques que vem crescendo no mundo são as Inteligências Artificiais. Estas são capazes de fazer coisas que, aos nossos olhos, parece ser complexo e inatingível a computadores, mas a tecnologia nos surpreende cada vez mais.

Com essa visão este trabalho tem como objetivo construir uma pseudo IA, onde ela somente terá uma espécie de "filtro", e não parâmetros mais complexos como *Machine Learning* por exemplo. Esta "IA" será um medidor de felicidade em reviews, onde mais próximo ao 100 por cento significa um texto extremamente positivo e mais próximo ao zero por cento, um texto extremamente negativo.

Para desenvolver este trabalho será utilizado a ferramenta Flex.

1.1. Sobre o Flex

O Flex (juntamente ao Bison que não será utilizado nesse trabalho) é uma ferramenta utilizada para entradas bem estruturadas. Ela foi feita originalmente para construir compiladores, contudo hoje é utilizada em diversas áreas [1]. Como nossa necessidade se encaixa perfeitamente à ferramenta, ela será utilizada.

2. Suposições para o código

Para o bom funcionamento do código e resultados esperados, teremos que fazer algumas considerações:

- O código foi construindo visando textos mais objetivos e sem abstrações, ou seja, poesias ou textos artísticos como entrada podem gerar um resultado não coerente;
- O texto de entrada deve estar na língua inglesa e no formato .txt;

- O código possui um dicionário de palavras positivas e negativas, ou seja, uma palavra só será contabilizada se ela existir nesse dicionário, caso contrário ela será considerada como uma palavra comum

3. O código

A ideia geral do código é bem simples: ocorrerá uma varredura por todo o texto dado na procura de palavras chave. Quando é encontrada uma palavra chave positiva, é incrementado em um na variável *positive*, assim, ao final da varredura, teremos o total de palavras chave positivas encontradas. O mesmo se aplica para palavras chave negativas.

Uma vez percorrido todo o texto será somado todas as palavras chave encontradas. Como o resultado final será a porcentagem de palavras positivas temos:

$$\text{resultado} = \frac{()palavrasPositivas}{totalPalavras} * 100$$

3.1. Primeira seção: Declarações

Nas declarações temos três variáveis: duas inteiras que serão os contadores de palavras chave e uma de ponto flutuante que será utilizada para realizar cálculos mais adiante. Logo após também é declarado o dicionário de palavras positivas e negativas. São estas palavras que, se encontradas no texto, serão contabilizadas ao contador

```
%{
/* TRABALHO 1 - Analisador léxico que pondera os sentimentos em um texto
   Onde: próximo de 0 é ruim e próximo de 1 é excelente
   Este analisador se aplica a qualquer tipo de review (filmes, jogos, etc.)
   Desenvolvido para a ferramenta Flex*/

int positive = 0, negative = 0;
float result = 0;
}%

POSITIVE {adorable|amazing|approve|awesome|accepted|acclaimed|admire|agreeable|attractive|brilliant|beautiful|cool|cute|
NEGATIVE {abysmal|adverse|alarming|annoy|atrocious|awful|bad|banal|barbed|boring|broken|clumsy|collapse|confused|corrupt
```

Figura 1. Print da seção 1 do código

3.2. Segunda seção: Lista de regras

Essa segunda seção é bem simples: caso encontrada algumas dessas palavras chave, é incrementado um ao contador. Nada mais.

```
%%  
{POSITIVE} {positive++;}  
{NEGATIVE} {negative++;}  
+  
%%
```

Figura 2. Print da seção 2 do código

3.3. Terceira seção: código C

O código C também não fará algo complexo. Em primeiro lugar ele lerá o arquivo de entrada dado na hora da execução do programa. Caso ele não encontre será adotada a entrada padrão - nesse caso o teclado. Logo após isso e a execução no analisador léxico já explicado (pelo comando `yylex()`) será calculado o total de palavras contadas e, por último, será calculado e mostrado a porcentagem de palavras positivas.

```
%%  
  
int main(argc, argv)  
{  
    int argc;  
    char **argv;  
    if (argc > 1){  
        if (!yyin = fopen(argv[1], "r")){  
            perror(argv[1]);  
            return(1);  
        }  
    }  
    yylex();  
    result = positive + negative;  
    printf("Result: %f per cent\n", result = (float)((positive)/result)*100);  
    return 0;  
}
```

Figura 3. Print da seção 3 do código

4. Resultado

Como já dito o algoritmo retorna uma porcentagem de felicidade no texto, ou seja, quanto mais próximo de 100 por cento, mais "feliz" o texto é.

Para o teste foi retirado uma review da internet e colocado em um arquivo .txt com o nome *text.txt*.

Para compilar deve-se seguir os seguintes passos:

```
flex projeto-int.l
```

```
gcc lex.yy.c -ll
```

```
./a.out text.txt
```

Após essas execuções, o algoritmo retornou com sucesso a porcentagem.

```
Result: 69.491524 per cent
```

Figura 4. Resultado após a análise

5. Conclusão

Este analisador léxico é bem simples comparado aos reais algoritmos de inteligência artificial. De fato o analisador não é o melhor caminho para uma real aplicação, porém a ferramenta Flex se mostra bem poderosa e útil para montagem de regras.

Referências

- [1] LEVINE, John R. flex & bison. 2009. O'Reilly: Sebastopol. 1. ed. pag. 1