



# **Correção ortográfica utilizando o algoritmo de *Viterbi***

**Ciência da Computação**

**Rafael Hengen Ribeiro, Régis Thiago Feyh, Ricardo Parizotto**

*{rafaelhr.ribeiro, registhiagofeyh, ricardo.dparizotto}@gmail.com*

***Inteligência Artificial***

***Prof. Dr. José Carlos Bins Filho***

## **1. Resumo**

O algoritmo de *Viterbi* é um algoritmo de programação dinâmica para encontrar a melhor sequência de estados ocultos - chamados de caminho de Viterbi - que resulta em uma sequência de eventos observados, especialmente no contexto das cadeias de *Markov* ocultas (*Hidden Markov Chains*).

## **2. Estruturas de Dados**

Arquivos são utilizados para armazenar o dicionário de palavras e as probabilidades de transições das letras.

As probabilidades de transições das letras são calculadas para todas as letras do alfabeto da língua portuguesa, eliminando diacríticos. São armazenadas em um espaço bidimensional da classe *Inicio*:

```

class Inicio    {
    private:
        ...

    public:
        map<string, int> mapWords;

        double transition[27][27];
        ....
}

```

A estrutura *mapWords* é uma árvore que armazena as palavras lidas no dicionário e a quantidade de vezes que ela aparece, permitindo que seja calculado as probabilidades de transição de letras, que são armazenadas em *transition*.

O valor das probabilidades de erro são armazenados na estrutura

```
static const double probTecla[27][27]
```

permitindo acessar diretamente a probabilidade de teclar um caractere 'x' tentando digitar um caractere 'z'.

## 2.1. Viterbi

O algoritmo de *Viterbi* é implementado percorrendo a evidencia (palavra dada pelo usuário) e calculando se as possíveis transições (de uma letra para outra) são mais prováveis, dadas as probabilidades calculadas com as palavras do dicionário, do que a sequencia de caracteres da evidencia:

$$ps[i][j] = pa[posMtr(c)] * ini.transition[posMtr(c)][i] * probTecla[i][posMtr(c)];$$

Onde:

- *ps* é um espaço bidimensional auxiliar utilizado para armazenar as iterações;
- *pa* é um espaço unidimensional que armazena as probabilidades anteriores;
- *ini.transition* armazena as probabilidades de transição;
- *probTecla* armazena as probabilidades de erro;

### 3. Funcionamento

O programa recebe uma palavra e testa se ela está no dicionário. Se ela não estiver, é chamado o algoritmo de *Viterbi* para calcular qual a palavra o usuário tentou digitar. Então, é feito o cálculo de distância de palavras de *Levenshtein* com as palavras do dicionário para descobrir qual a palavra mais próxima da obtida.

```
Entrada: zxha
-----
acha >> Viterbi
acha >> menor distancia c/ viterbi
```

Figura 1. Exemplo de utilização.

Na Figura 1. o usuário inseriu a palavra “zxha”, que não está no dicionário. O algoritmo de Viterbi encontrou “acha”. E o algoritmo de menor distância também encontrou “acha”, que está no dicionário.

### REFERÊNCIAS

- [1] FORNEY JR, G. David. The viterbi algorithm. **Proceedings of the IEEE**, v. 61, n. 3, p. 268-278, 1973.
- [2] LOU, Hui-Ling. Implementing the Viterbi algorithm. **Signal Processing Magazine, IEEE**, v. 12, n. 5, p. 42-52, 1995.
- [3] A Gentle Introduction to Dynamic Programming and the Viterbi Algorithm, <[http://www.cambridge.org/resources/0521882672/7934\\_kaeslin\\_dynpro\\_new.pdf](http://www.cambridge.org/resources/0521882672/7934_kaeslin_dynpro_new.pdf)>, **acesso em 1 de junho de 2016**.
- [4] Distance Between Strings, <[http://www.cut-the-knot.org/do\\_you\\_know/Strings.shtml](http://www.cut-the-knot.org/do_you_know/Strings.shtml)>, **acesso em 3 de junho de 2016**.